

TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN COMPUTACIÓN

Herramienta automática para el análisis y clasificación de imágenes de capilaroscopia del lecho ungueal

Estudiante: Julia Gil Ordóñez
Directora: María Noelia Barreira Rodríguez

A Coruña, 3 de septiembre de 2019.

A todos aquellos que me dieron una palmadita en la espalda cuando más lo necesitaba.

Agradecimientos

Me gustaría agradecer a mi familia y amigos por no quejarse y aguantar todas esas horas de trabajo en las que ponía música de los 80 a todo volumen. También dar gracias a todos esos profesores que me ayudaron a continuar con todo y a no dejar nunca de ser curiosa. Y en especial, gracias a Noelia por guiarme y darme buenos consejos para seguir adelante con este proyecto.

Resumen

Las capilaroscopias son técnicas de diagnóstico vascular no invasivas. Se aplican sobre el lecho ungueal y toman una imagen microscópica de la zona, mostrando los vasos sanguíneos que allí se encuentran. Con ellas se pueden detectar enfermedades como lupus eritomatoso, dermatomiositis o, con mayor frecuencia, esclerodermia sistémica. En esta última, suele presentarse una alteración vascular llamada fenómeno de Raynaud, que afecta directamente a las arterias de pequeño calibre, en especial a las de los dedos de las manos o los pies.

El proceso de diagnóstico es largo y tedioso: para un diagnóstico correcto, se han de tomar imágenes de todos los dedos y cada una de ellas debe ser analizada por un experto. Entonces, éste detecta manualmente los capilares visibles y los clasifica uno a uno.

El objetivo de este proyecto es el de agilizar este proceso y proporcionar a los expertos una herramienta automática que detecte los capilares en las imágenes, los cuente y los clasifique, realizando un diagnóstico fiable y rápido. Se desarrollará también una aplicación web para facilitar el uso de la metodología propuesta a un público más general.

Palabras clave:

- Vaso sanguíneo
- Capilaroscopia
- Red de neuronas artificiales
- CNN
- U-Net
- Diagnóstico
- Raynaud
- Esclerodermia sistémica
- Lecho ungueal
- Segmentación

Índice general

1	Introducción	1
1.1	Esclerodermia sistémica	1
1.2	Realización de la capilaroscopia	2
1.3	Estado del arte	3
1.4	Objetivos	4
1.5	Estructura de la memoria	4
2	Metodología de desarrollo y planificación	7
2.1	Metodología	7
2.1.1	Actividades	7
2.1.2	Roles	8
2.1.3	Herramientas	8
2.1.4	Adaptación de la metodología	9
2.2	Planificación	9
2.3	Seguimiento	10
2.4	Estimación de costes	11
3	Tecnologías y materiales	13
3.1	Banco de imágenes	13
3.2	Lenguaje de programación y librerías	14
3.3	Entornos de desarrollo	15
3.4	Edición de imágenes	15
3.5	Control de versiones	16
3.6	Gestión de proyectos	16
3.7	Equipo de trabajo	16
4	Fundamentos teóricos	17
4.1	Redes de Neuronas Artificiales	17

4.2	Redes de Neuronas Convolucionales (CNN)	19
4.3	U-Net	21
5	Metodología	23
5.1	Preprocesado	23
5.2	Segmentación de imágenes	24
5.2.1	Entrenamiento con imágenes originales	26
5.2.2	Entrenamiento con canal rojo	27
5.2.3	Entrenamiento con canal azul	28
5.2.4	Entrenamiento con imágenes en escala de grises	29
5.2.5	Entrenamiento con imágenes de 512×512 píxeles	30
5.2.6	Resultados	30
5.3	Conteo de capilares	34
5.3.1	Corrección de orientación	34
5.3.2	Extracción de capilares	34
5.3.3	Resultados	36
5.4	Clasificación	37
5.4.1	Obtención de la longitud	38
5.4.2	Obtención del ancho	39
5.4.3	Regularidad según variabilidad de orientaciones del gradiente	41
5.4.4	Regularidad según área de la <i>bounding box</i>	42
5.4.5	Resultados	43
6	Aplicación web	47
6.1	Análisis	47
6.2	Diseño	55
6.2.1	Maquetación	55
6.2.2	Patrones de diseño	58
6.2.3	Base de Datos	61
6.2.4	Arquitectura	61
6.3	Implementación	63
6.4	Pruebas	72
7	Conclusiones	73
A	Glosario de acrónimos	77
B	Glosario de términos	79

Bibliografía

81

Índice de figuras

1.1	Ejemplo de un paciente con síndrome de Raynaud.	2
1.2	Distinción entre capilar sano (a) y enfermo (b).	3
2.1	Funcionamiento de Scrum.	8
2.2	Diagrama de Gantt de la planificación del proyecto.	10
2.3	Diagrama de Gantt del seguimiento del proyecto.	11
3.1	Ejemplo de imágenes que se descartarán. (a) Imagen con burbujas, (b) Imagen con reflejos, (c) Imagen con hemorragia.	13
3.2	Resultado del marcado manual. (a) Imagen original, (b) Máscara realizada de forma manual.	14
4.1	Esquema de la estructura de una RNA.	18
4.2	Esquema de la estructura de una CNN.	21
4.3	Segmentación de una imagen y clasificación de los segmentos mediante U-Net.	21
4.4	Arquitectura de la U-Net.	22
5.1	(a) Imagen original. (b) Imagen con contraste aumentado.	24
5.2	(a) Imagen antes del filtrado. (b) Imagen después del filtrado Gaussiano.	24
5.3	Ejemplo de modo de relleno <i>nearest</i>	25
5.4	Imágenes resultantes tras aplicar modificadores a una imagen de capilaroscopia.	26
5.5	Imagen original.	27
5.6	Imagen del canal rojo.	28
5.7	Imagen del canal azul.	29
5.8	Imagen en escala de grises.	29
5.9	Resultado del marcado de la U-Net. (a) Máscara resultante, (b) Máscara con la resolución original restablecida.	31
5.10	Curva ROC de los distintos modelos de entrenamiento seleccionados.	32

5.11	Esquema de los pasos del proceso de conteo.	34
5.12	(a) Imagen orientada hacia abajo. (b) Imagen orientada hacia arriba.	35
5.13	Contornos detectados.	35
5.14	Contornos detectados después de realizar el filtrado por perímetro.	35
5.15	Imagen con los extremos recortados.	36
5.16	(a) Capilares pertenecientes a un solo contorno. (b) Contorno recortado para poder separar los capilares.	36
5.17	Ejemplo de detección de capilares. Fila superior: Imágenes originales. Fila inferior: Imágenes correspondientes segmentadas con los capilares detectados encuadrados.	37
5.18	Ejemplo de detección fallida de capilares. Fila superior: Imágenes originales. Fila inferior: Imágenes correspondientes segmentadas con los capilares detectados encuadrados.	38
5.19	Cálculo de longitud del capilar.	39
5.20	Ejemplo de capilar anómalo largo (marcado en azul).	39
5.21	(a) Contorno entero. (b) Contorno recortado al 70 %.	40
5.22	Centroide de la imagen.	41
5.23	Recuento de píxeles blancos de la fila del centroide.	41
5.24	(a) Porcentaje obtenido en capilar irregular (41.96 %). (b) Porcentaje obtenido en capilar regular (72.70 %).	43
5.25	Diferencias de longitudes en los capilares. Capilar rodeado más pequeño: 129 píxeles de longitud. Capilar rodeado más largo: 306 píxeles de longitud.	44
5.26	Capilar tortuoso con una regularidad bastante alta (40.61 %) usando el método de cálculo de orientaciones del gradiente.	44
5.27	(a) Porcentaje de regularidad obtenido con el método de la <i>bounding box</i> (58.47 %). (b) Porcentaje de regularidad obtenido con el método mixto (54.45 %).	45
5.28	Ejemplo de clasificación errónea de capilares.	46
5.29	Ejemplo de clasificación correcta de capilares.	46
6.1	Casos de uso detectados para este proyecto.	48
6.2	Pantalla de inicio de sesión.	55
6.3	Pantalla de registro de un usuario nuevo.	56
6.4	Pantalla de ver, buscar y registrar pacientes.	56
6.5	Pantalla de registro de un paciente nuevo.	57
6.6	Pantalla de perfil de un paciente.	58
6.7	Pantalla de diagnóstico de una capilaroscopia.	59
6.8	Pantalla de perfil de usuario.	59
6.9	Esquema de Modelo-Vista-Controlador.	60

6.10	Esquema de patrón Instancia única.	61
6.11	Diagrama Entidad-Relación.	62
6.12	Diagrama de clases de la aplicación.	62
6.13	Diagrama de la clase UnetModelSingleton	63
6.14	Diagrama de secuencia para diagnosticar una imagen.	64
6.15	Diagrama del módulo views.py	65
6.16	Pantalla de inicio de sesión.	66
6.17	Pantalla de registro de un usuario nuevo.	67
6.18	Pantalla de inicio.	67
6.19	Pantalla de registro de pacientes.	68
6.20	Parte superior del perfil del paciente.	68
6.21	Parte inferior del perfil del paciente, donde se ven las capilaroscopias.	69
6.22	Parte superior de la pantalla de diagnóstico.	69
6.23	Parte inferior de la pantalla de diagnóstico.	70
6.24	Pantalla de perfil de usuario.	70
6.25	Pantalla de modificar el perfil de usuario.	71
6.26	Pantalla de modificar la contraseña del usuario.	71
6.27	Tests efectuados para el modelo de la aplicación.	72

Índice de tablas

2.1	Tabla de coste estimado por <i>sprint</i>	12
2.2	Tabla de coste total estimado del proyecto.	12
2.3	Tabla de coste real por <i>sprint</i>	12
2.4	Tabla de coste total real del proyecto.	12
5.1	Parámetros de modificación de las imágenes	25
5.2	Parámetros de entrenamiento de la red.	27
5.3	Parámetros de entrenamiento de la red con en canal rojo.	28
5.4	Parámetros de entrenamiento de la red con el canal azul.	28
5.5	Parámetros de entrenamiento de la red con imágenes en escala de grises. . . .	30
5.6	Parámetros de entrenamiento de la red con imágenes de 512×512 píxeles. . .	30
5.7	Resultados del entrenamiento con variaciones en el conjunto de entrenamien- to usado.	31
5.8	Métricas estadísticas obtenidas en el modelo entrenado con las imágenes ori- ginales, para cada umbral.	33
5.9	Valores para determinar el mejor umbral.	33
5.10	Valores para determinar la clasificación.	45
6.1	CU-01: Registrarse.	49
6.2	CU-02: Identificarse.	49
6.3	CU-03: Ver pacientes.	50
6.4	CU-04: Ver perfil.	50
6.5	CU-05: Registrar paciente.	51
6.6	CU-06: Buscar paciente.	51
6.7	CU-07: Ver perfil paciente.	52
6.8	CU-08: Modificar perfil.	53
6.9	CU-09: Subir capilaroscopias paciente.	53
6.10	CU-10: Modificar perfil paciente.	54

6.11 CU-11: Ver diagnósticos paciente.	54
--	----

Introducción

Una capilaroscopia es una técnica de diagnóstico que estudia los capilares sanguíneos de manera económica y no invasiva. Se aplica sobre el área de las uñas, obteniendo imágenes microscópicas de los vasos que allí se encuentran. Con las imágenes resultantes, se pueden observar ciertos factores en los capilares que podrían determinar alguna enfermedad de tipo sistémico. Con mayor frecuencia se utiliza para el diagnóstico diferencial de la **esclerodermia sistémica**, aunque también podrían detectar otras afecciones como dermatomiositis, síndrome de Sjögren o lupus eritematoso [1].

1.1 Esclerodermia sistémica

La esclerodermia (o esclerosis) sistémica es una enfermedad crónica poco común caracterizada por cambios degenerativos y fibrosis en la piel, las articulaciones y los órganos internos, así como por alteraciones en los vasos sanguíneos [2]. Esta enfermedad es cuatro veces más frecuente en las mujeres que en los hombres y suele presentarse entre los 20 y los 50 años de edad. La alteración vascular más frecuente es el fenómeno de Raynaud, en el cual los dedos de las manos se vuelven de manera repentina y temporal muy pálidos y presentan hormigueo o quedan entumecidos, doloridos, o ambas cosas. Esto suele darse como respuesta al frío o a un disgusto emocional.

El fenómeno o síndrome de Raynaud es una arteriopatía periférica funcional en la que las arterias de pequeño calibre (arteriolas), en particular las de los dedos de las manos o de los pies, se contraen con más fuerza [3]. Puede presentarse sin causa aparente (primario) o como síntoma de alguna afección (secundario). El síndrome de Raynaud primario es mucho más frecuente que el secundario. Entre un 60% y un 90% de los casos de síndrome de Raynaud primario se presentan en mujeres de edades comprendidas entre los 15 y los 40 años. Aunque también se hace uso de ecografías Doppler [4] para diagnosticar esta enfermedad, la capilaroscopia posee la ventaja de permitir establecer su etiología [5][6].

En la figura 1.1 se puede ver una imagen de un paciente con síndrome de Raynaud. Las áreas blancas (palidez) son resultado de la ausencia de flujo sanguíneo debida al estrechamiento de los vasos sanguíneos. Las áreas blancas se desarrollan irregularmente en los dedos [7].



Figura 1.1: Ejemplo de un paciente con síndrome de Raynaud.

1.2 Realización de la capilaroscopia

Los aparatos que realizan las capilaroscopias se denominan **capilaroscopios**. Contienen una lente que se posiciona sobre la uña del paciente y, con un haz luz que lo ilumina, toman la imagen del lecho ungueal.

Para que la imagen resultante sea correcta se debe recomendar al paciente las siguientes acciones [8]:

- No utilizar esmalte de uñas.
- No fumar dos horas antes de la realización de la prueba.
- Permanecer en una habitación con una temperatura media de 22-25°, para evitar que los capilares se contraigan a causa de factores externos.

Se debe tomar una imagen de cada uno de los dedos, a menos que en éstos haya algún tipo de alteración (hemorragias, infecciones...).

Una vez realizadas las capilaroscopias, se podrán observar los principales parámetros que determinarán si los capilares son sanos o presentan algún tipo de anomalía. Esos parámetros son [9]:

- La longitud. Generalmente los vasos sanos son largos.
- El diámetro. Los vasos sanos suelen ser finos.

- Su densidad por mm^2 . La cantidad en las personas adultas sanas oscila entre 10 y 17 capilares por mm^2 .
- Algún tipo de anomalía morfológica. Los capilares enfermos presentan irregularidades, como tortuosidades, neovascularización, etc.

En la figura 1.2 se puede observar la diferencia entre capilares sanos y anómalos.



Figura 1.2: Distinción entre capilar sano (a) y enfermo (b).

1.3 Estado del arte

Actualmente existen varios trabajos referentes a las capilaroscopias que ayudan a automatizar el diagnóstico. En 2016, Berks et al. [10] desarrollaron un sistema que aúna varias de las funcionalidades anteriormente descritas, no siendo capaz de realizar un diagnóstico con la suficiente precisión (entre un 74% y un 84% en la detección de los capilares). Estos mismos crearon en 2018 una herramienta que examina el flujo sanguíneo y las formas de los capilares en las capilaroscopias mediante vídeos en alta definición del lecho ungueal [11] haciendo uso del software de diagnosis que habían desarrollado. Aunque mejoraron los porcentajes de detección en un 10%, el proceso de análisis para cada capilaroscopia era lento (entre uno y dos minutos).

Por otra parte, investigadores de la Thomas Jefferson University han creado un algoritmo que, marcando dónde se sitúan los capilares y dónde el fondo, se cuentan de manera automática [12]. Esto agiliza bastante el proceso de diagnóstico, aunque no evita que el experto tenga que examinar cada una de las capilaroscopias para marcarlas. Este contratiempo lo consiguen paliar investigadores de la Wroclaw Medical University, que han desarrollado software que los localiza, aunque no los cuenta ni los clasifica [13].

Por tanto, a pesar de que existen en la literatura algunas herramientas para el análisis de imágenes de capilaroscopia, éstas son incompletas, lentas, requieren la interacción del usuario o no tienen una precisión suficiente.

1.4 Objetivos

El objetivo de este trabajo es desarrollar una metodología que realice un diagnóstico basándose en capilaroscopias. Primero, se desarrollará un método para la detección automática de los capilares sanguíneos en las imágenes mediante aprendizaje automático. Luego, se diseñará un algoritmo que realice el conteo y el análisis de la morfología de los vasos para, finalmente, poder clasificarlos en *capilares sanos* y *capilares anómalos*. Para integrar y facilitar el uso de este proyecto a un público más general, se creará una aplicación web que implemente la metodología desarrollada. A esta aplicación se le enviarán las imágenes que se tomen de los pacientes. De esta forma, los expertos sólo tendrían que enviar las imágenes a la aplicación y ésta les informaría del posible diagnóstico. La aplicación también actuará como un gestor de pacientes: cada experto podrá añadir pacientes nuevos, modificar sus datos y guardarlos junto con las imágenes tomadas de cada uno de ellos en una base de datos.

1.5 Estructura de la memoria

A continuación se detallará la estructura de esta memoria, con una breve descripción del contenido que se encontrará en cada capítulo:

- **Capítulo 2: Metodología de desarrollo y planificación.** Se dará una breve explicación de la metodología de desarrollo elegida para la realización de este proyecto, seguida de una exposición de la planificación y posterior seguimiento. También se incluirá una estimación de costes.
- **Capítulo 3: Tecnologías y materiales.** Se enumerarán las tecnologías utilizadas para llevar a cabo todo el trabajo, como por ejemplo el lenguaje de programación elegido o sus librerías.
- **Capítulo 4: Fundamentos teóricos.** Se hará un resumen de las técnicas que se van a utilizar para la segmentación de los capilares.
- **Capítulo 5: Metodología.** Se entrará en detalle en la descripción de la metodología propuesta: el preprocesado de las capilaroscopias, la segmentación que se ha efectuado, los criterios para obtener el conteo de los capilares y cómo se ha diseñado el algoritmo de clasificación.

- **Capítulo 6: Aplicación web.** Se describirá el análisis, diseño, la implementación y las pruebas de la aplicación web.
- **Capítulo 7: Conclusiones y trabajos futuros.** En este capítulo se plasmarán las conclusiones y posibles trabajos futuros.

Metodología de desarrollo y planificación

En este capítulo se describirá la metodología de desarrollo elegida y se describirá la planificación realizada en base a ella, así como el seguimiento y la estimación de costes del proyecto.

2.1 Metodología

La metodología con la que se ha llevado a cabo el proyecto es Scrum. Scrum es un marco de trabajo para el desarrollo iterativo e incremental. En él se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados a corto plazo y donde los requisitos son cambiantes o poco definidos [14].

2.1.1 Actividades

En Scrum un proyecto se ejecuta en ciclos temporales cortos llamados iteraciones o **sprints**. Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite. Estos son los procesos que se llevan a cabo en una iteración:

1. Planificación de la iteración (*Sprint Planning*).
2. Ejecución de la iteración (*Sprint*).
3. Reunión diaria de sincronización del equipo (*Scrum Daily Meeting*).
4. Demostración de los requisitos completados (*Sprint Review*).

5. Retrospectiva (*Sprint Retrospective*).
6. Refinamiento de la lista de requisitos y cambios en el proyecto.

En la Figura 2.1 se muestra un esquema del funcionamiento de Scrum.

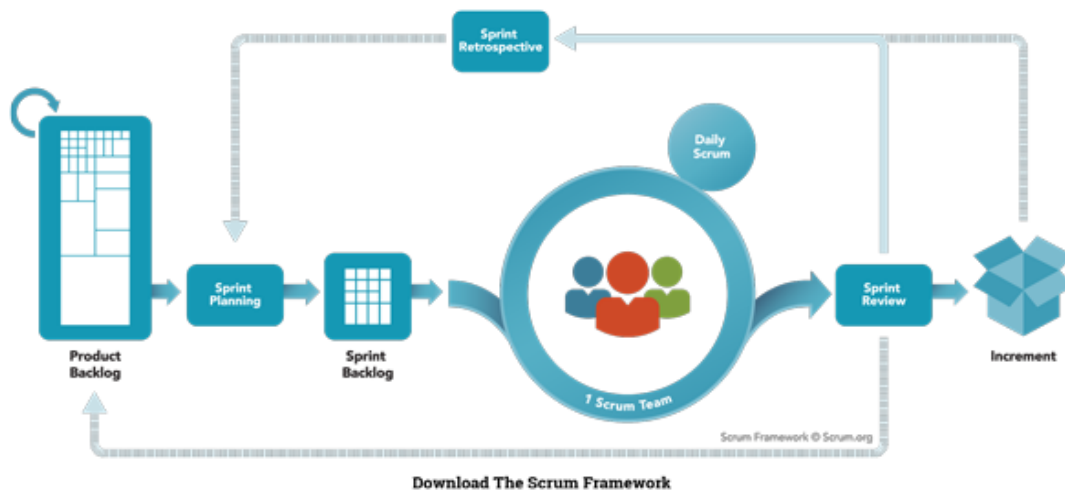


Figura 2.1: Funcionamiento de Scrum.

2.1.2 Roles

Existen tres roles diferentes en Scrum:

- **Cliente** (*Product Owner*): Su principal misión es encargarse de que exista una priorización clara de los objetivos a conseguir, con el propósito de maximizar el valor del trabajo que lleva a cabo el equipo. Colaborará con el equipo para planificar, revisar y dar detalle a los objetivos de cada iteración.
- **Equipo de desarrollo**: Son los que desarrollan el producto. Crean los incrementos resultantes de cada iteración.
- **Facilitador** (*Scrum Master*): Se encarga de facilitar reuniones entre el equipo y el cliente. Es el responsable de gestionar y asegurar el seguimiento del proceso Scrum.

2.1.3 Herramientas

Estas son las herramientas que se usan en el desarrollo de Scrum:

- **Lista de requisitos priorizada** (*Product Backlog*). Contiene los objetivos/requisitos de alto nivel del producto o proyecto, que se suelen expresar en forma de historias de

usuario. Para cada objetivo/requisito se indica el valor que aporta al cliente y el coste estimado de completarlo. La lista está priorizada balanceando el valor que cada requisito aporta al negocio frente al coste estimado que tiene su desarrollo.

- **Lista de tareas de la iteración** (*Sprint Backlog*). Subconjunto de objetivos/requisitos del Product Backlog seleccionado para la iteración actual y su plan de tareas de desarrollo.
- **Gráficos de trabajo pendiente** (*Burndown Chart*). Un gráfico de trabajo pendiente a lo largo del tiempo muestra la velocidad a la que se está completando los objetivos/requisitos. Permite extrapolar si el Equipo podrá completar el trabajo en el tiempo estimado.

2.1.4 Adaptación de la metodología

Puesto que el equipo que conforma el proyecto no se adapta al designado por Scrum, se hará una adaptación de la metodología para nuestro caso.

El equipo de este proyecto está formado por la directora del TFG y una alumna. La directora hará a la vez de cliente y de facilitadora: definirá los requisitos y ayudará a la alumna en la realización del proyecto. La alumna se corresponderá con el equipo de desarrollo, que diseñará e implementará el producto bajo la supervisión de la directora.

También se harán cambios en el proceso de cada iteración: la planificación se llevará a cabo en reuniones entre la alumna y la directora, donde se comentarán los objetivos a cumplir en el *sprint*. De no ser posible la reunión, se mantendrá el contacto vía correo electrónico. Lo mismo se aplica a la demostración de requisitos completados y al refinamiento de la lista de requisitos y cambios en el proyecto. La reunión diaria de sincronización del equipo no se efectuará, ya que el equipo se compone de una única persona. La alumna hará un balance al final de cada jornada, sopesando lo que ha realizado y planificando los objetivos del día siguiente.

Cuando una tarea requiera tiempo de procesamiento y no sea necesaria la atención completa de la alumna, se permitirá la realización de *sprints* en paralelo.

2.2 Planificación

En esta sección se detallará la planificación establecida para el proyecto. En la Figura 2.2 se muestra un diagrama de Gantt, donde se indican las fechas de inicio y fin de cada uno de los *sprints* planificados. Se han establecido siete *sprints* diferentes:

1. **Creación de máscaras manuales.** En este *sprint* se filtrarán las imágenes de calidad insuficiente (bajo contraste, borrosas, con artefactos, ...) de la base de datos y se crearán

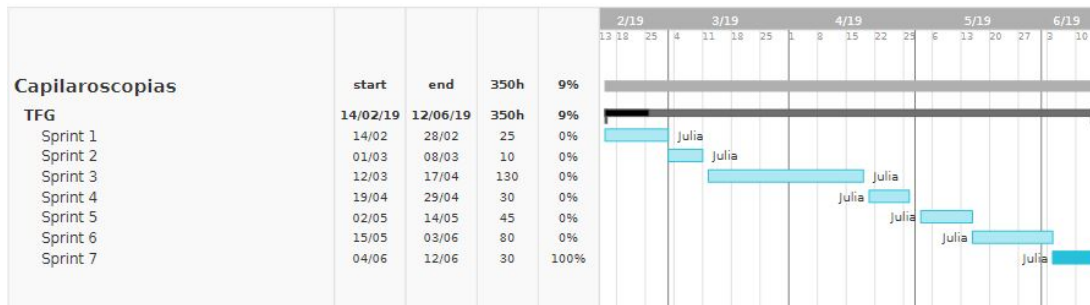


Figura 2.2: Diagrama de Gantt de la planificación del proyecto.

máscaras binarias que indicarán la localización de los vasos sanguíneos en las capilaroscopias.

2. **Preprocesado de imágenes.** En este *sprint* se adecuarán las imágenes de la base de datos para que los capilares sean más fácilmente detectables.
3. **Segmentación de imágenes.** El objetivo de este *sprint* será el de conseguir un modelo de segmentación de imágenes que logre identificar los vasos sanguíneos en las capilaroscopias con una precisión suficiente.
4. **Desarrollo de una metodología para el conteo de capilares.** En este *sprint* se creará una metodología para hacer recuento de cuántos capilares existen en cada una de las capilaroscopias.
5. **Desarrollo de una metodología para la clasificación de capilares.** Se creará una metodología que clasifique los capilares detectados en cada imagen como *sano* o *anómalo*.
6. **Creación de la aplicación web.** Este *sprint* está dividido en cuatro tareas: el análisis, el diseño, la implementación y las pruebas de la aplicación web que integrará la metodología de conteo y clasificación.
7. **Redacción de la memoria.**

2.3 Seguimiento

A continuación se detallará el seguimiento efectuado al finalizar el proyecto. Debido a exámenes y prácticas se ha tenido que retrasar la realización de los *sprints* del 3 en adelante hasta después de Junio, lo que ha variado significativamente el calendario original de la planificación. En la Figura 2.3 se puede observar el diagrama de Gantt perteneciente al seguimiento.

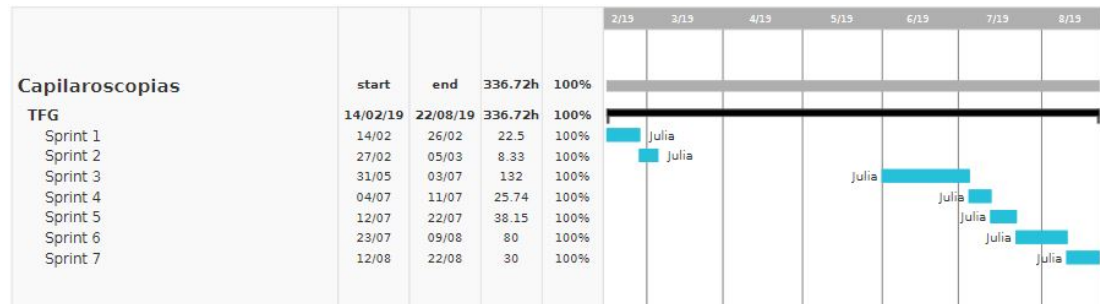


Figura 2.3: Diagrama de Gantt del seguimiento del proyecto.

El *sprint* 3 finalmente se demoró en 2 horas, aunque hubo varios *sprints* que se realizaron en menos tiempo del esperado: los *sprints* 1 y 2 se han realizado en 2 horas menos de lo planeado y el *sprint* 5 también pasó de durar 45 horas a 38.

2.4 Estimación de costes

En este apartado se mostrará la estimación de los costes para el desarrollo del proyecto. Se han considerado cinco roles diferentes: Desarrollador novel, Analista, Diseñador, Investigador y Jefe de proyecto. Los costes del trabajo de cada uno de ellos, de media, son los siguientes:

- Desarrollador novel: 15 €/hora
- Analista: 30 €/hora
- Diseñador: 25 €/hora
- Investigador: 20 €/hora
- Jefe de proyecto: 40 €/hora

En el caso de este proyecto, los cuatro primeros roles se considerarán uno solo (Desarrollador, Analista, Diseñador e Investigador), que cobrará 20 €/hora, trabajando 5 horas al día. Se ha supuesto que el Jefe de proyecto ha dedicado 35 horas a la supervisión de este trabajo. En la Tabla 2.1 se puede ver el desglose de coste estimado por cada *sprint* y en la Tabla 2.2 el coste estimado total.

En la tabla 2.3 se muestra el desglose de costes reales por *sprint* y en la Tabla 2.4 el coste total real del proyecto.

Tabla 2.1: Tabla de coste estimado por *sprint*.

<i>sprint</i>	Horas	Coste
1- Creación de máscaras manuales	25	500 €
2- Preprocesado de imágenes	10	200 €
3- Segmentación de imágenes	130	2.600€
4- Desarrollo de una metodología para el conteo de capilares	30	600 €
5- Desarrollo de una metodología para la clasificación de capilares	45	900 €
6- Creación de la aplicación web	80	1.600 €
7- Redacción de la memoria	30	600 €
Total		7.000 €

Tabla 2.2: Tabla de coste total estimado del proyecto.

Perfil	Coste
Desarrollador / Analista / Diseñador / Investigador	7.000 €
Jefe de proyecto	1.200 €
Total	8.200 €

Tabla 2.3: Tabla de coste real por *sprint*.

<i>sprint</i>	Horas	Coste
1- Creación de máscaras manuales	22.5	450 €
2- Preprocesado de imágenes	8.3	166 €
3- Segmentación de imágenes	132	2.640€
4- Desarrollo de una metodología para el conteo de capilares	25.7	514 €
5- Desarrollo de una metodología para la clasificación de capilares	40	800 €
6- Creación de la aplicación web	80	1.600 €
7- Redacción de la memoria	30	600 €
Total		6.770 €

Tabla 2.4: Tabla de coste total real del proyecto.

Perfil	Coste
Desarrollador / Analista / Diseñador / Investigador	6.770 €
Jefe de proyecto	1.200 €
Total	7.970 €

Tecnologías y materiales

Para la realización de este proyecto se utilizarán una serie de materiales y herramientas que se describirán brevemente a continuación.

3.1 Banco de imágenes

Se dispone de una base de datos con 92 imágenes de capilaroscopias con una resolución de 1280×1024 píxeles, entre las que se hará una criba. Se descartarán aquellas que sean difusas o presenten algún tipo de elemento que dificulte la correcta visualización de los vasos sanguíneos. Estos elementos podrían ser burbujas, reflejos o hemorragias severas (Figura 3.1). Luego, se procederá a procesarlas para que los capilares sanguíneos se aprecien mejor a simple vista.

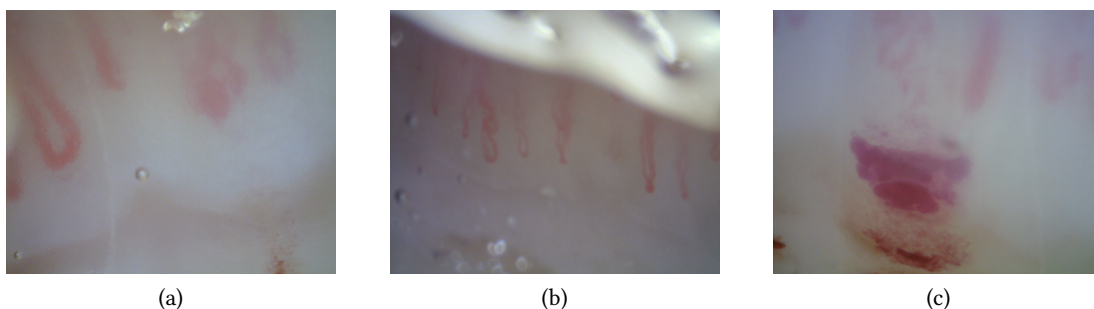


Figura 3.1: Ejemplo de imágenes que se descartarán. (a) Imagen con burbujas, (b) Imagen con reflejos, (c) Imagen con hemorragia.

Una vez realizado el proceso de cribado, se reduce el número de imágenes a 52 capilaroscopias válidas.

A continuación se procede a un marcado manual de las imágenes. El objetivo de este paso es la creación de imágenes en blanco y negro que indiquen dónde se encuentran los capilares

en cada imagen. Se generará así un conjunto de máscaras binarias que servirán en el proceso de entrenamiento y validación del sistema. Para la realización del marcado manual se hace uso de un programa de edición de imagen. Se colorea de blanco las zonas que corresponden a capilares y se deja el resto de la imagen en negro (Figura 3.2).

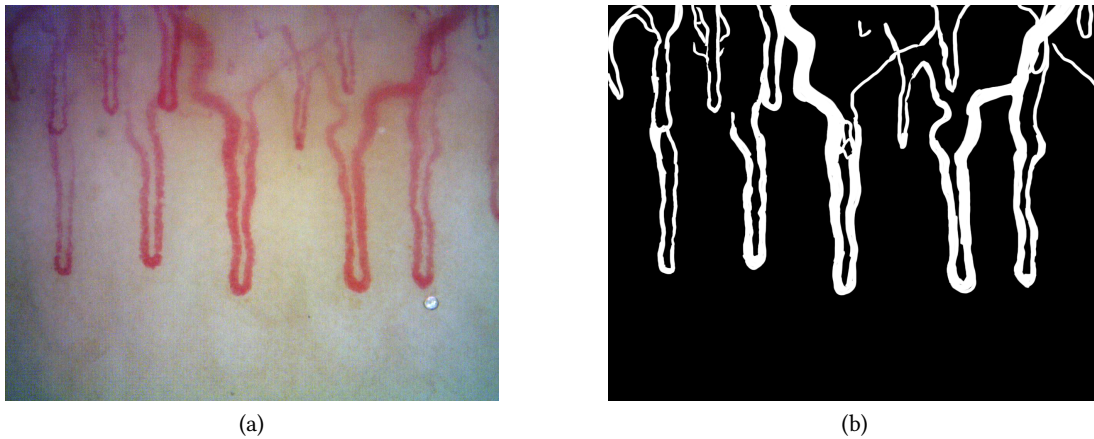


Figura 3.2: Resultado del marcado manual. (a) Imagen original, (b) Máscara realizada de forma manual.

3.2 Lenguaje de programación y librerías

Python es un lenguaje de programación multiplataforma interpretado y de alto nivel. Permite la programación de diferentes paradigmas: imperativo, orientado a objetos y funcional. Su documentación es amplia y de alta calidad, lo que permite un aprendizaje rápido y una adaptabilidad sencilla. Se usa tanto para desarrollo de software, como para desarrollo web (parte servidor), programas matemáticos y scripting de sistemas [15].

Las principales librerías de python que se han utilizado en el desarrollo de este trabajo son las siguientes:

- **Numpy:** Numpy es una librería de código abierto de Python enfocada a operaciones de computación científica. Añade operaciones con matrices, vectores y álgebra lineal [16].
- **Python Imaging Library (PIL):** Esta librería aporta un extenso soporte para formatos de imagen en Python, así como una representación eficiente de ellos y también herramientas de procesamiento [17].
- **TensorFlow:** TensorFlow es una librería de código abierto para aprendizaje máquina y computación numérica desarrollada por Google. Permite distribuir cálculos entre CPU y GPU e incluso paralelizarlos para que se realicen simultáneamente [18].

- **Keras:** Keras es una API para la implementación de redes neuronales de alto nivel, escrita en Python y capaz de correr sobre TensorFlow, CNTK o Theano. Permite un prototipado rápido y sencillo y además soporta redes convolucionales y recurrentes.
- **OpenCV:** OpenCV es una librería de código abierto y multiplataforma que implementa funciones de visión artificial, tales como filtros, funciones de umbralización o detectores de bordes. Está optimizada para funcionar en aplicaciones a tiempo real [19].
- **Scikit-Learn:** Scikit-Learn es una librería de aprendizaje máquina de código abierto. Implementa funciones de clasificación, regresión, *clustering* y reducción de dimensionalidad, entre otras [20].
- **Django:** Django es un *framework* de alto nivel para Python que permite un desarrollo rápido de entornos web. Es de código abierto, rápido, seguro y escalable [21].

Para la gestión de la base de datos de la aplicación web se utilizará **SQLite**. SQLite es una librería para la gestión de bases de datos de tipo relacional. No precisa de un servidor para funcionar, es transaccional y tampoco es necesaria su configuración. Es de código abierto [22].

3.3 Entornos de desarrollo

Para la creación del código fuente del trabajo y la elaboración de diagramas se utilizarán los siguientes programas:

- **Spyder:** Spyder es un entorno de desarrollo orientado a la programación científica en Python. Está escrito en Python e implementa herramientas para la instalación sencilla de librerías. Permite la visualización de gráficas y además posee terminal y depurador propios [23].
- **draw.io:** draw.io es un entorno de diseño de diagramas online gratuito [24]. Provee herramientas para la creación de una gran variedad de diagramas: UML, BPMN, de flujo, de Venn, etc.

3.4 Edición de imágenes

Para la creación de las máscaras binarias se ha utilizado el programa **Adobe Photoshop**. Adobe Photoshop es un programa de diseño gráfico desarrollado por Adobe que permite el tratamiento de imágenes de forma digital, tanto en 2D como en 3D [25].

3.5 Control de versiones

Para el control de versiones del trabajo se ha utilizado **Github**. GitHub es una plataforma para alojar proyectos que utiliza el sistema de control de versiones Git. Su código fuente está escrito en Ruby on Rails [26].

3.6 Gestión de proyectos

Para la gestión de este proyecto se ha usado **Taiga**. Taiga es una aplicación web para la gestión de proyectos gratuita y de código abierto. Permite la gestión de varias metodologías, como Scrum o Kanban [27].

3.7 Equipo de trabajo

El equipo en el que se realizará este proyecto tiene las siguientes características:

- Modelo: Hewlett-Packard HP ENVY 15 Notebook PC
- Sistema Operativo: Windows 10 Home Edition
- Procesador: Intel Core i7-4702MQ 2.20GHz
- Número de núcleos: 8
- Tarjeta gráfica: NVIDIA GeForce GT 750M 4GB DDR3
- Memoria RAM: 16 GB

Debido a que el equipo anterior no poseía suficiente potencia para realizar ciertos experimentos en el entrenamiento de redes de *deep learning*, también se utilizó otro equipo con las siguientes características:

- Modelo: Lenovo Legion Y520
- Sistema Operativo: Windows 10 Home Edition
- Procesador: Intel Core i7 7^a generación 2.20GHz
- Número de núcleos: 8
- Tarjeta gráfica: NVIDIA GeForce GTX 1050 DDR4 8GB
- Memoria RAM: 16 GB

Fundamentos teóricos

EN este apartado se describirán los fundamentos teóricos de las principales técnicas que se utilizarán para llevar a cabo el proyecto.

4.1 Redes de Neuronas Artificiales

Las redes de neuronas artificiales (RNA) son una de las herramientas principales en el aprendizaje máquina. Su diseño está inspirado en la fisiología del cerebro humano, por lo que intentan imitar su forma de procesar la información. Así como las neuronas biológicas son procesadores simples que transforman una serie de entradas y las envían a otras neuronas, una neurona artificial es también un elemento que, en base a un conjunto de entradas x , produce una única salida y y la propaga.

Los elementos más importantes de una neurona artificial son los siguientes [28]:

- Las entradas que reciben los datos (x_j). En una neurona biológica corresponderían a las dendritas
- Los pesos sinápticos w_{ij} . En las neuronas biológicas, el intercambio de información se produce en la sinapsis. En ella, las dendritas de una neurona se comunican con el axón de las siguientes. En una neurona artificial a cada entrada que proviene de otras neuronas se les asigna un peso, es decir, un factor de importancia para dicha entrada.
- Una regla de propagación. Es una operación que se realiza con la entrada y que dará como resultado lo que se denomina potencial postsináptico ($h_i(t)$). Una de las operaciones más comunes es sumar las entradas de forma ponderada según el peso asociado a cada una, como se muestra en la siguiente ecuación:

$$h_i(t) = \sum w_{ij}x_j \quad (4.1)$$

La otra regla de propagación más habitual es la distancia euclídea entre la entrada y su peso asociado:

$$h_i(t) = \sum (x_j - w_{ij})^2 \quad (4.2)$$

- Una función de activación. El valor obtenido con la regla de propagación se filtra a través de una función conocida como función de activación y es la que da la salida de la neurona.

En general, las neuronas suelen agruparse en unidades estructurales llamadas capas. Dentro de una capa, las neuronas suelen ser del mismo tipo. Se pueden distinguir tres tipos de capas:

- Capa de entrada: reciben datos o señales procedentes del entorno.
- Capa de salida: proporcionan la respuesta de la red a los estímulos de la entrada.
- Capas ocultas: se corresponden al procesamiento interno de la red. Estas capas pueden variar en número e incluso omitirse por completo.

Aunque existen varios tipos de redes de neuronas artificiales, uno de los más comunes es el perceptrón multicapa, cuya estructura se puede apreciar en la Figura 4.1

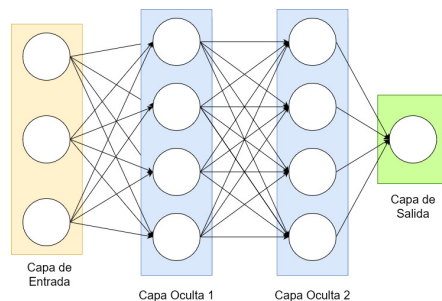


Figura 4.1: Esquema de la estructura de una RNA.

Para conseguir que la red de neuronas aprenda deberá ser entrenada, es decir, sometida a un proceso de actualización de los pesos w_{ij} . En este proceso existen tres conceptos fundamentales:

- Paradigma de aprendizaje: información de la que dispone la red. Puede ser supervisado o no supervisado. En el paradigma supervisado, la red conoce la salida deseada, por lo que intenta minimizar el error entre ésta y la calculada por ella misma. En el no supervisado la red no conoce la respuesta deseada, por lo que debe extraer características o agrupar patrones similares.

- Regla de aprendizaje: principios que gobiernan el aprendizaje. Pueden ser de cuatro tipos: Boltzmann, Hebb, Competitivo o de Minimización del error.
- Algoritmo de aprendizaje: procedimiento numérico de ajuste de los pesos.

El entrenamiento requiere de varias iteraciones en las cuales todas las muestras se envían a través de la red. Cada iteración se divide en varios pasos: primero se propagan los valores de la entrada hacia la salida, atravesando todas las capas intermedias. Una vez que se obtienen los resultados de la capa de salida, se calcula su proximidad al valor deseado (en caso de paradigmas supervisados). Comúnmente se busca la minimización del error y para calcularla se utiliza una función de pérdida. Dependiendo del problema al que se haga frente, se elegirá la función de pérdida más adecuada: entropía cruzada, error cuadrático medio, error medio absoluto, etc.

Cuando se ha hallado la pérdida, se produce una retropropagación del error desde la salida hacia la entrada. Se irán modificando los pesos usando el algoritmo de aprendizaje elegido (por ejemplo, el gradiente descendente).

Los principales usos para las redes de neuronas artificiales son:

- Problemas de clasificación.
- Problemas de regresión.
- Ajuste de curvas.
- Agrupación de elementos similares.
- Reducción de dimensionalidad.
- Visualización de datos de representación compleja.

4.2 Redes de Neuronas Convolucionales (CNN)

Una red de neuronas convolucional (CNN) es un algoritmo de *deep learning* que toma como entrada una imagen, asigna importancia a los elementos que la componen a través los pesos y, de esta forma, es capaz de clasificar el contenido de las imágenes [29]. El preprocesado en una CNN es mucho menor que en un algoritmo de clasificación convencional, puesto que en estos últimos la extracción de características se realiza manualmente, mientras que las CNN, con el suficiente entrenamiento, son capaces de aprenderlas por sí mismas.

Su arquitectura está inspirada en la organización de la corteza visual humana. Las neuronas individuales responden a los estímulos sólo si están situadas en una región específica del campo visual, llamada el campo receptivo. Entonces, para cubrir toda el área de visualización, se solapan varios campos receptivos.

Como su nombre indica, su funcionamiento se basa en convoluciones. Una convolución es un proceso mediante el cual a una imagen se le aplica un filtro a través de operaciones matriciales [30]. Se puede definir matemáticamente como la matriz generada a partir de una matriz $F_{m \times n}$ y una matriz $G_{(2N+1) \times (2N+1)}$ llamada *kernel* o núcleo, con $2N + 1 < m, n$, siguiendo la siguiente ecuación:

$$(F * G)(i, j) = \sum_{s=-a}^a \sum_{t=-b}^b f(i + s, j + t)g(s, t) \quad (4.3)$$

con $i = a \dots M - a$ y $j = b \dots N - b$

Los valores del kernel serán considerados los pesos en la red de neuronas.

En general, las redes neuronales convolucionales se organizan en tres tipos de capas distintas [31]:

- Una capa convolucional. Aquí es donde se realiza la convolución y da como resultado un mapa de las características de la imagen original. De esta forma se reduce el tamaño de los parámetros. En una misma red puede haber varias capas de convolución.
- Una capa de reducción (*pooling*). Se coloca generalmente después de la capa convolucional. Su utilidad principal radica en la reducción de las dimensiones espaciales de la entrada para la siguiente capa convolucional. La reducción de tamaño conduce también a la pérdida de información, aunque este aspecto puede ser beneficioso para la red; la disminución en el tamaño de la entrada conlleva una menor sobrecarga de cálculo para las próximas capas.
- Una capa de clasificación. Al final de todas las posibles capas convolucionales y de reducción, las redes utilizan generalmente capas completamente conectadas. Su estructura es igual a la de las redes de neuronas convencionales en las que cada píxel de la imagen será considerado como una neurona individual. Esta capa tendrá tantas neuronas como el número de clases que se deban predecir.

Esta estructura se puede apreciar en la Figura 4.2 [32]. La entrada de la red es una imagen, mientras que a salida es la clase a la que pertenece dicha entrada.

Los problemas para los que se suele usar una red convolucional son los siguientes:

- Detección/categorización de objetos en imágenes.
- Clasificación de escenas.
- Clasificación de imágenes en general.

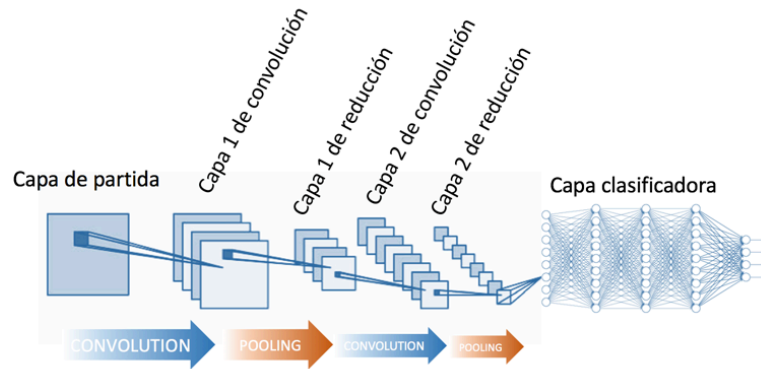


Figura 4.2: Esquema de la estructura de una CNN.

4.3 U-Net

La U-Net es un tipo de red convolucional. Está considerada como una de las arquitecturas estándar para la clasificación de imágenes donde no solo se necesita definir la imagen en su conjunto, sino también segmentar partes de ella y etiquetarlas. A diferencia de las CNN, la U-Net devuelve una imagen con los resultados de la segmentación (Figura 4.3) [33].



Figura 4.3: Segmentación de una imagen y clasificación de los segmentos mediante U-Net.

La arquitectura de la U-Net (Figura 4.4) es una combinación de convoluciones y convoluciones traspuestas. Estas últimas tienen un funcionamiento muy similar a las ordinarias, excepto que el kernel que usan en lugar de reducir las dimensiones de las muestras, las amplía. Gracias a esto se unen en una sola imagen las predicciones obtenidas en los procesos de clasificación de las diferentes partes. El proceso se compone de dos capas simétricas:

1. Capa de bajada o de contracción: Es una sucesión de convoluciones y *pooling* que va reduciendo el tamaño de la imagen.

2. Capa de subida o de expansión: Devuelve el tamaño original de la imagen con convoluciones traspuestas que usan la imagen del mismo nivel de la capa de bajada para que no se pierdan detalles.

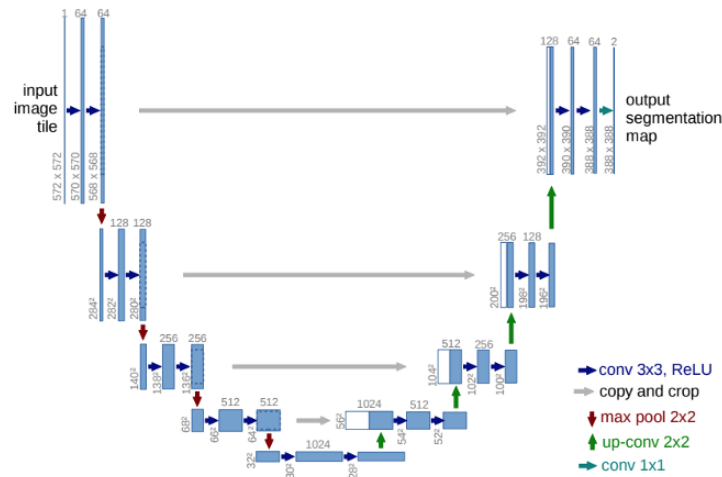


Figura 4.4: Arquitectura de la U-Net.

La U-Net es muy usada en problemas de segmentación biomédica.

Capítulo 5

Metodología

En este capítulo se describirá la metodología desarrollada para la detección, conteo y clasificación de capilares, así como las pruebas que se han efectuado y los resultados obtenidos en cada una de ellas.

5.1 Preprocesado

Para facilitar la localización de los vasos en las imágenes disponibles, se llevará a cabo un preprocesado. Primero se aumenta el contraste de cada una. Para realizar este proceso se ejecuta una modificación del histograma siguiendo la siguiente fórmula:

$$ImagenContraste(x, y) = \alpha I(x, y) + \beta \quad (5.1)$$

donde α se corresponde con el factor de corrección de contraste, β con el factor de corrección de brillo e $I(x, y)$ con el píxel que se está modificando [34].

Se intentará imitar el proceso de contraste de los programas de edición fotográfica, por lo que será necesario variar un tanto esta fórmula para que se adecue a la que se utiliza en ellos [35]:

$$ImagenContraste = f(I(x, y) - 127) + 127 = fI(x, y) + 127(1 - f) \quad (5.2)$$

donde f es el factor de corrección, que se calcula de la siguiente forma:

$$f = \frac{259(C + 255)}{255(259 - C)} \quad (5.3)$$

siendo C el grado de contraste que se desea aplicar.

Se han hecho experimentos con distintos valores para C y en el que mejor se aprecian los capilares sin distorsionar la imagen es en el rango de 60 a 65, concretamente en $C = 64$.

En la Figura 5.1 se puede apreciar la diferencia de contraste antes y después de aplicar las modificaciones.

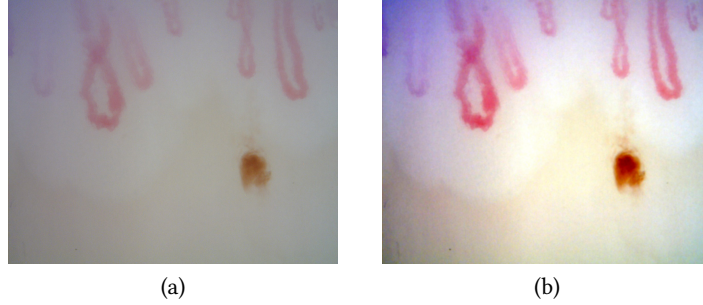


Figura 5.1: (a) Imagen original. (b) Imagen con contraste aumentado.

Se ha detectado que, al aumentar el contraste, en muchas de las imágenes aparecía ruido pixelado, probablemente de aplicar el capilaroscopia sobre el aceite para realizar la fotografía. Para suavizarlo se han filtrado las imágenes con un filtro Gaussiano. Este es un tipo de filtro de paso bajo, que reduce los componentes de alta frecuencia y que da como resultado una imagen más difuminada. El *kernel* que se utiliza para el filtro Gaussiano debe ser impar y sus valores dependen de un parámetro σ , que será el que indique el grado de filtrado. Cada elemento de la matriz se calcula de la siguiente manera:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \quad (5.4)$$

Para el problema que se plantea en este apartado, se ha elegido un *kernel* de 7×7 , con un $\sigma = 2$. En la Figura 5.2 se puede observar la diferencia al aplicar este filtro.

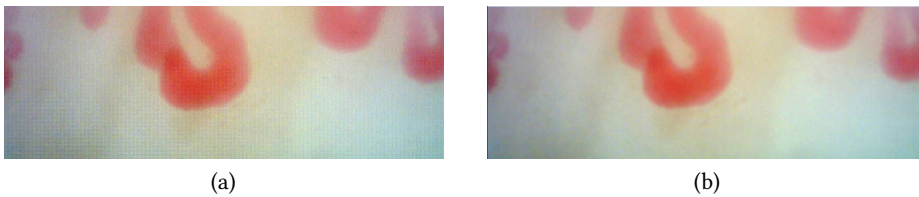


Figura 5.2: (a) Imagen antes del filtrado. (b) Imagen después del filtrado Gaussiano.

5.2 Segmentación de imágenes

La segmentación de imágenes se realizará usando una red U-Net. Para entrenar dicha red, se dividen las 52 imágenes de las que disponemos en dos grupos: entrenamiento y test. En este caso se han elegido 30 para el conjunto de entrenamiento y 22 para el de test.

En primer lugar, para incrementar la diversidad de los datos de entrada se utilizan técnicas de aumento de datos (*data augmentation*). Así, se generan más imágenes de entrenamiento a partir de las originales. Se rotan, se reflejan y se amplían siguiendo unos valores predeterminados, que son los que se muestran en la Tabla 5.1. El nombre es bastante descriptivo en todos los casos, a excepción quizás del parámetro *Modo de relleno*. Éste se refiere a qué valores se asignarán cuando sea necesario conocer la intensidad de píxeles que están fuera de los límites de la imagen de entrada original. Por ejemplo, cuando se haga una reducción de anchura en la entrada y se desee mantener la resolución original, los bordes que quedaran vacíos, se rellenarán siguiendo el método especificado en este parámetro. En nuestro caso se usará *nearest*, que indica que se tomará el valor del píxel más cercano en la imagen original para realizar el relleno. Se puede apreciar un ejemplo de este modo en la Figura 5.3.

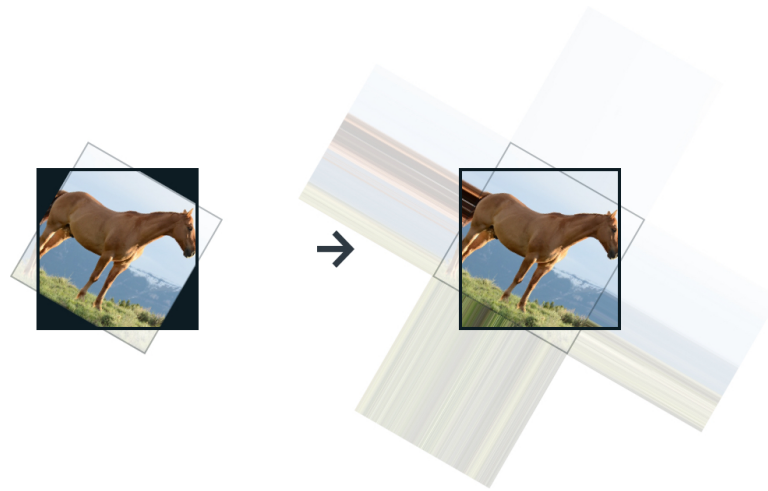


Figura 5.3: Ejemplo de modo de relleno *nearest*.

Tabla 5.1: Parámetros de modificación de las imágenes

Modificador	Valor
<i>Rango de rotación</i>	50 %
<i>Rango de cambio de anchura</i>	5 %
<i>Rango de cambio de altura</i>	5 %
<i>Rango de recorte</i>	5 %
<i>Rango de zoom</i>	5 %
<i>Volteo horizontal</i>	True
<i>Modo de relleno</i>	'nearest'

Esto se aplica tanto a las capilaroscopias como a sus máscaras y los resultados son los que aparecen en la figura 5.4.

Se han probado distintos tipos de conjuntos de entrenamiento: se ha intentado segmentar con imágenes en escala de gris, aislando el canal rojo de las muestras, aislando el canal azul y con distintos tamaños de imagen. En las siguientes subsecciones se analizará cada uno de ellos

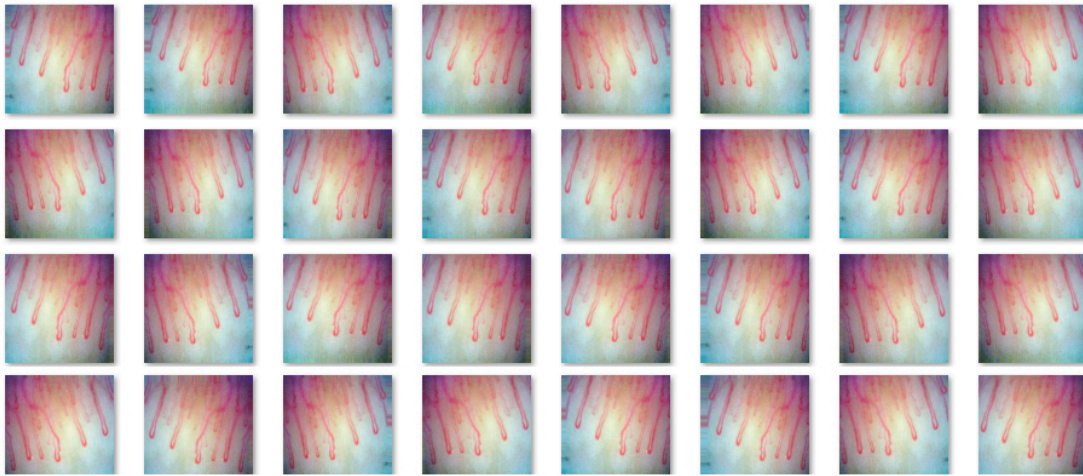


Figura 5.4: Imágenes resultantes tras aplicar modificadores a una imagen de capilaroscopia.

en detalle. Además, se analizará el impacto de los siguientes parámetros de entrenamiento en los resultados del sistema:

- **Tamaño de lote** (*batch size*): El número de elementos de entrenamiento que se pasarán simultáneamente a la red para que se propaguen hacia delante y hacia atrás. Cuanto mayor sea el tamaño del lote, más memoria se necesitará para ejecutar el entrenamiento, aunque también mejorará la precisión de la estimación del gradiente.
- **Iteraciones**: El número de veces que se propagan hacia delante y hacia atrás todos los elementos del conjunto de entrenamiento.
- **Pasos por iteración**: El número total de pasos (lotes de elementos) que se procesarán antes de determinar que una iteración ha terminado.
- **Tamaño de objetivo**: La resolución en píxeles que tendrán las imágenes de salida.
- **Espacio de color de la imagen**: El espacio de color de las imágenes de salida.

La función de pérdida (*loss*) que utiliza la U-Net es la de *entropía cruzada* [36] y para calcular la precisión (*accuracy*) se hace un porcentaje de cuántos valores predichos por la red son correctos, comparándolos con la máscara manual correspondiente a la imagen de entrada.

5.2.1 Entrenamiento con imágenes originales

En este primer experimento, la entrada a la U-Net está formada por las imágenes originales preprocesadas (Figura 5.5).

Los parámetros con los que se entrenó la red son los mostrados en la Tabla 5.2:



Figura 5.5: Imagen original.

Tabla 5.2: Parámetros de entrenamiento de la red.

Parámetro	Valor
<i>Tamaño de lote</i>	2
<i>Iteraciones</i>	2
<i>Pasos por iteración</i>	300
<i>Tamaño de objetivo</i>	(256,256)
<i>Espacio de color</i>	'rgb'

Como resultado del entrenamiento se ha obtenido una *accuracy* del 93.63 % y un porcentaje de *loss* del 15.74 %. Se han modificado los parámetros de la siguiente manera para intentar mejorar estos porcentajes:

- Se ha aumentado el tamaño de lote a 3, pero esto aumentaba demasiado el consumo de memoria y no había diferencia en los resultados.
- Se ha realizado una sola iteración, pero las imágenes producidas en el test resultaban ser imágenes totalmente grises debido a que la red necesitaba más información para realizar el entrenamiento. También se aumentó el número de iteraciones a 3, pero los porcentajes obtenidos no mejoraron con respecto a la configuración original.
- Se aumentaron los pasos por iteración a 600, pero los resultados no se veían afectados. De la misma forma disminuyeron a 150 y el resultado fue el mismo que al reducir el número de iteraciones: las imágenes resultantes del test eran totalmente grises.

5.2.2 Entrenamiento con canal rojo

En este caso, las imágenes con las que trabaja la red son las resultantes de eliminar los canales verde y azul a las imágenes originales (Figura 5.6). Se ha probado este método ya que en los capilares se muestran con un tono más rojizo que el resto de la imagen. De esta forma, tal vez la U-Net sería capaz de una mejor identificación de las estructuras de interés.

Los parámetros de entrenamiento son los mostrados en la Tabla 5.3.



Figura 5.6: Imagen del canal rojo.

Tabla 5.3: Parámetros de entrenamiento de la red con en canal rojo.

Parámetro	Valor
<i>Tamaño de lote</i>	2
<i>Iteraciones</i>	2
<i>Pasos por iteración</i>	300
<i>Tamaño de objetivo</i>	(256,256)
<i>Espacio de color</i>	'rgb'

En el entrenamiento se ha obtenido una *accuracy* del 86.19 % y un porcentaje de *loss* del 38.46 %. Se puede observar (tanto en la Figura 5.6 como en los resultados) que la localización de los vasos en las imágenes es menos evidente que en el caso de las capilaroscopias originales preprocesadas. Al igual que con las imágenes originales, se ha intentado mejorar los porcentajes resultantes modificando los parámetros de entrenamiento:

- Se ha aumentado el número de iteraciones, así como el número de pasos por iteración, aunque los resultados no se vieron afectados.

5.2.3 Entrenamiento con canal azul

Se ha probado también a entrenar la red eligiendo solamente el canal azul en las imágenes. Donde hay vasos sanguíneos, el color azul es el menos abundante; a causa de esto, si sólo se selecciona el canal azul de las imágenes, los capilares aparecerán resaltados en negro (Figura 5.7). De esta forma se podrían identificar con mayor facilidad.

Los parámetros para el entrenamiento de la red son los mostrados en la Tabla 5.4.

Tabla 5.4: Parámetros de entrenamiento de la red con el canal azul.

Parámetro	Valor
<i>Tamaño de lote</i>	2
<i>Iteraciones</i>	2
<i>Pasos por iteración</i>	300
<i>Tamaño de objetivo</i>	(256,256)
<i>Espacio de color</i>	'rgb'

Se ha obtenido una *accuracy* del 90.57 % y una *loss* del 23.97 %. Se ha mejorado con respecto



Figura 5.7: Imagen del canal azul.

al canal rojo, aunque sigue sin superar al entrenamiento con las imágenes originales. Se han realizado las mismas modificaciones de parámetros que para el canal rojo y tampoco se ha obtenido un resultado que mejorara la configuración original.

5.2.4 Entrenamiento con imágenes en escala de grises

Se ha entrenado la red con las capilaroscopias en escala de grises. Para conseguirlas se ha aplicado la transformada de luma ITU-R 601-2, que sigue la siguiente ecuación [37][38]:

$$L = R \frac{299}{1000} + G \frac{587}{1000} + B \frac{114}{1000} \quad (5.5)$$

donde R , G y B son los valores de los canales rojo, verde y azul respectivamente y L es la imagen resultante en escala de grises.

Los vasos sanguíneos se aprecian con bastante claridad en este tipo de imágenes (Figura 5.8).



Figura 5.8: Imagen en escala de grises.

Los parámetros con los que se ha entrenado la red son los mostrados en la Tabla 5.5.

En el proceso de entrenamiento se ha obtenido una *accuracy* de 86.19 % y una *loss* del 67.73 %. Es un nivel de pérdida demasiado alto para ser aceptable.

Tabla 5.5: Parámetros de entrenamiento de la red con imágenes en escala de grises.

Parámetro	Valor
<i>Tamaño de lote</i>	2
<i>Iteraciones</i>	2
<i>Pasos por iteración</i>	300
<i>Tamaño de objetivo</i>	(256,256)
<i>Espacio de color</i>	'grayscale'

5.2.5 Entrenamiento con imágenes de 512×512 píxeles

Como se ha observado que los mejores resultados de entrenamiento se han obtenido con las imágenes originales preprocesadas, por lo que se aumentó la resolución con la que trabaja la U-Net a 512×512 píxeles. De este modo, se podrían mejorar los detalles de segmentación. Los parámetros para el entrenamiento de la red son los mostrados en la Tabla 5.6.

Tabla 5.6: Parámetros de entrenamiento de la red con imágenes de 512×512 píxeles.

Parámetro	Valor
<i>Tamaño de lote</i>	1
<i>Iteraciones</i>	2
<i>Pasos por iteración</i>	600
<i>Tamaño de objetivo</i>	(512,512)
<i>Espacio de color</i>	'rgb'

En el entrenamiento se ha obtenido una *accuracy* del 93.45 % y una *loss* de 18.23 %. Se puede ver que mejora la *accuracy* de la resolución original, aunque no el porcentaje de *loss*. Se incrementó el tiempo que tarda la red en realizar el proceso de test con respecto a la resolución de 256×256 píxeles, pasando de 15 segundos a 40 (un incremento de 1.4 segundos por imagen). En este experimento se realizó el entrenamiento con un tamaño de lote de 1 a causa de las limitaciones técnicas del ordenador en el que se ejecutó. Se probó también a realizar el entrenamiento con 300 pasos por iteración, pero el resultado no mejoraba.

5.2.6 Resultados

Los resultados tras el entrenamiento de estos experimentos se resumen en la Tabla 5.7. En ella se incluyen los porcentajes de *accuracy* y de *loss*. Se ve que la U-Net obtiene una mayor *accuracy* (93.68 %) usando la configuración original. Como se ha dicho anteriormente, las imágenes que recibe tienen una resolución de 1280×1024 píxeles, aunque la U-Net las reduce para poder trabajar con ellas de manera eficiente. Una vez obtenida la imagen resultado, esta se reescala hasta su tamaño original (Figura 5.9).

A continuación se procede a realizar la validación de los resultados del entrenamiento a través de las 22 imágenes seleccionadas para test. Para ello, se van a utilizar las métricas estadísticas de sensibilidad, especificidad, *accuracy* y exactitud. Para poder calcular estas métricas

Tabla 5.7: Resultados del entrenamiento con variaciones en el conjunto de entrenamiento usado.

Método	Loss	Accuracy
Imágenes originales	15.74 %	93.63 %
Eligiendo canal azul en las imágenes	23.97 %	90.57 %
Eligiendo canal rojo en las imágenes	38.46 %	86.19 %
Imágenes en escala de grises	67.73 %	86.19 %
Imágenes con un tamaño de 512×512 píxeles	18.23 %	93.45 %

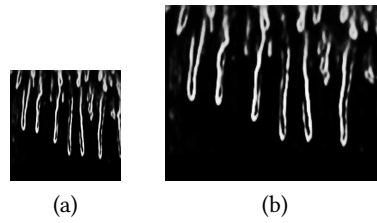


Figura 5.9: Resultado del marcado de la U-Net. (a) Máscara resultante, (b) Máscara con la resolución original restablecida.

se hace uso del recuento de Verdaderos Positivos, Verdaderos Negativos, Falsos Positivos y Falsos Negativos:

- **Verdaderos Positivos (VP):** Son aquellos píxeles que indican una zona de interés (en este caso un capilar) y que coinciden en la máscara manual y en la predicha por la red. Es decir, la red clasifica correctamente ese píxel como un capilar.
- **Verdaderos Negativos (VN):** Son aquellos píxeles que indican que no es una zona de interés (los correspondientes al fondo de la máscara) y que coinciden en la máscara manual y en la predicha por la red. Es decir, la red clasifica correctamente ese píxel como fondo.
- **Falsos Positivos (FP):** Son aquellos píxeles que en la máscara manual se marcan como fondo (negativos) y la red los clasifica como capilares (positivos).
- **Falsos Negativos (FN):** Son aquellos píxeles que en la máscara manual se marcan como positivos y la red los clasifica como negativos.

Una vez obtenidos estos datos, se procede a calcular las métricas de la siguiente forma:

$$\text{Sensibilidad} = \frac{VP}{VP + FN} \quad (5.6)$$

$$\text{Especificidad} = \frac{VN}{VN + FP} \quad (5.7)$$

$$Accuracy = \frac{VP}{VP + FP} \quad (5.8)$$

$$Exactitud = \frac{VP + VN}{VP + FP + FN + VN} \quad (5.9)$$

Se van a analizar los tres modelos con mayor porcentaje de *accuracy* en el entrenamiento para la validación. Estos son: el entrenado con las imágenes originales preprocesadas, el entrenado con canal azul de las capilaroscopias y el que hacía uso de las imágenes de 512×512 píxeles. Las imágenes devueltas por la red no son binarias, si no que están en escala de grises, por lo que para poder compararlas con las máscaras manuales, se les realiza una umbralización. Una umbralización es un proceso mediante el cual una imagen que posee varios tonos de gris pasa a ser binaria (es decir, blanca y negra). Para hacer esto, se establece un valor umbral. Se determinará el color del píxel siguiendo la siguiente ecuación:

$$I'(x, y) = \begin{cases} 0 & \text{si } I(x, y) > t \\ 255 & \text{si } I(x, y) \leq t \end{cases} \quad (5.10)$$

siendo t el valor umbral, I la imagen original, I' la imagen resultante tras la umbralización y x y y las coordenadas del píxel que se analiza.

Una vez obtenida la imagen umbralizada, se procede a calcular las métricas estadísticas descritas anteriormente.

Para establecer el mejor modelo se ha utilizado una curva ROC (Figura 5.10). Se ha analizado cada uno de los modelos con un rango de valores umbral entre 25 y 75, con intervalos de 5.

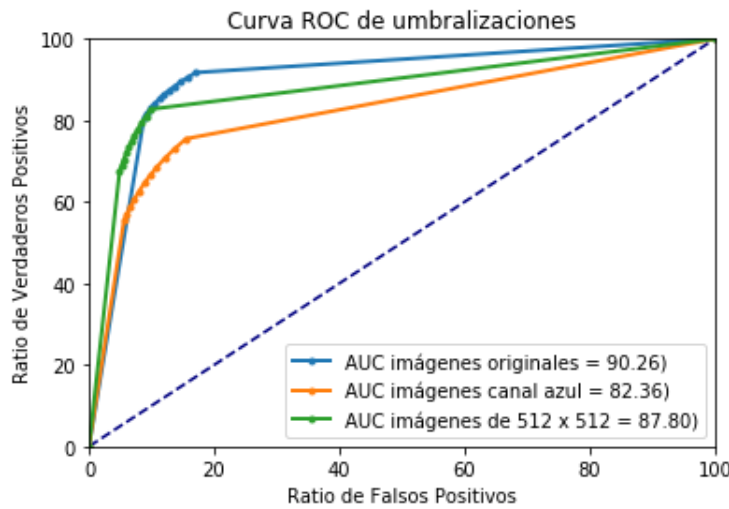


Figura 5.10: Curva ROC de los distintos modelos de entrenamiento seleccionados.

Una vez obtenidos los datos de la curva ROC, se puede decir que el mejor modelo es el entrenado utilizando las imágenes originales, puesto que es el que ha obtenido un área bajo la curva (AUC) mayor (90.26). Se analizarán, pues, los valores de las métricas que se han obtenido en ese modelo para cada uno de los umbrales (Tabla 5.8). También se añadirá la desviación típica de *accuracy*. Esta métrica indica la variación que hay en los porcentajes de *accuracy* de las imágenes. Cuanto más bajo sea este valor, más estable será la configuración.

En base a la curva ROC de este modelo concreto, se ha calculado cuál es el mejor umbral utilizando la siguiente expresión:

$$d = \sqrt{(1 - \text{sensibilidad})^2 + (1 - \text{especificidad})^2} \quad (5.11)$$

siendo el mejor resultado aquel que más se aproxime a 0. Los resultados obtenidos son los mostrados en la Tabla 5.9.

Tabla 5.8: Métricas estadísticas obtenidas en el modelo entrenado con las imágenes originales, para cada umbral.

Valor umbral	Sensibilidad	Especificidad	<i>Accuracy</i>	Exactitud	Desviación típica de <i>accuracy</i>
25	90.71 %	83.48 %	84.59 %	87.09 %	5.74 %
30	89.42 %	84.78 %	85.45 %	87.10 %	5.34 %
35	88.25 %	85.89 %	86.22 %	87.07 %	5.03 %
40	87.15 %	86.92 %	86.89 %	86.98 %	4.76 %
45	86.32 %	87.82 %	87.51 %	86.86 %	4.51 %
50	84.90 %	88.50 %	88.07 %	86.70 %	4.29 %
55	83.84 %	89.20 %	88.59 %	86.52 %	4.08 %
60	82.70 %	89.85 %	89.07 %	86.28 %	3.89 %
65	81.54 %	90.45 %	89.52 %	86.00 %	3.72 %
70	80.37 %	91.00 %	89.93 %	85.69 %	3.56 %
75	79.22 %	91.52 %	90.33 %	85.37 %	3.4 %

Tabla 5.9: Valores para determinar el mejor umbral.

Valor umbral	Resultado
25	18.95
30	18.54
35	18.36
40	18.34
45	18.32
50	18.98
55	19.44
60	20.06
65	20.78
70	21.59
75	22.44

El valor obtenido más próximo a 0 es el del umbral de 45 (18.32), por lo que se puede establecer como el umbral óptimo de este modelo.

5.3 Conteo de capilares

Una vez hallado el mejor modelo para la U-Net, se procederá a realizar el conteo de capilares en cada imagen, partiendo de las máscaras umbralizadas con el valor óptimo obtenido en el paso anterior. En la Figura 5.11 se muestra un esquema con los pasos que se seguirán en el proceso de conteo.

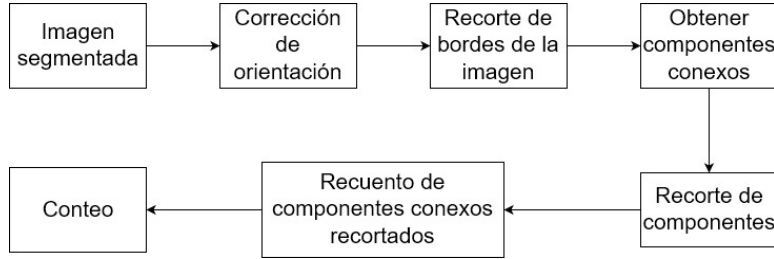


Figura 5.11: Esquema de los pasos del proceso de conteo.

5.3.1 Corrección de orientación

En primer lugar, se analiza la orientación de las imágenes. La mayoría de las capilaroscopias de la base de datos están orientadas de forma que los capilares desciendan desde arriba, aunque hay algunas volteadas verticalmente. A la hora de realizar el conteo esto podría ser problemático, por lo que se corregirán las imágenes para que todas estén orientadas de la misma forma.

Se puede ver en la Figura 5.12 que en las imágenes orientadas hacia abajo, el último cuarto de imagen está prácticamente en negro, mientras que en las orientadas hacia arriba es donde se concentra la mayor cantidad de píxeles blancos. Por esto, se hace un recuento de cuántos píxeles en blanco hay desde la esquina superior izquierda hasta un porcentaje κ del total de la imagen y también del $1 - \kappa$ restante. Si la concentración de píxeles blancos en la parte inferior es la mayor de ambas, la imagen se volteará.

5.3.2 Extracción de capilares

Tras la corrección de la orientación de las imágenes, se procede al conteo. El primer paso es obtener los contornos de los componentes conexos de la imagen. De esta forma, se podrán localizar los vasos sanguíneos, aunque también se resaltarán pequeñas manchas blancas por toda la imagen que corresponden a ruido resultante de la umbralización. Los contornos detectados son los encuadrados en la Figura 5.13.

Como se puede percibir, hay mucho ruido que se deberá descartar antes de poder realizar el conteo propiamente dicho. Para filtrar las manchas se calcula el perímetro de cada uno

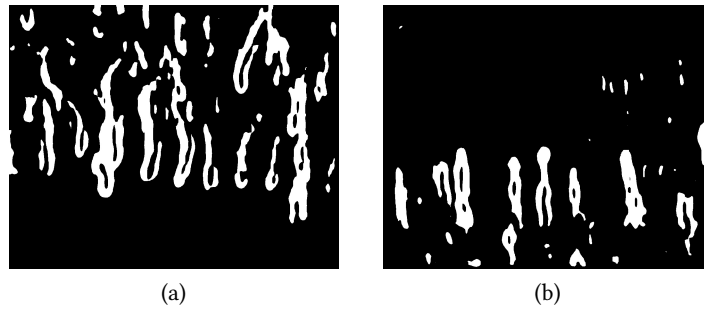


Figura 5.12: (a) Imagen orientada hacia abajo. (b) Imagen orientada hacia arriba.

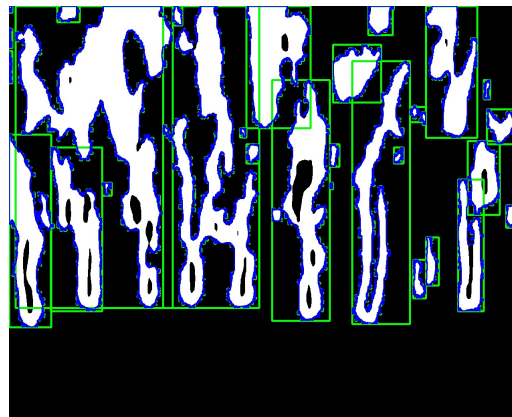


Figura 5.13: Contornos detectados.

de los contornos. Se establecerá un valor v para el umbral de perímetro válido. De esta forma, aquellas formas que no lo superen, no se tendrán en consideración. En la Figura 5.14 se muestra la imagen tras el filtrado.

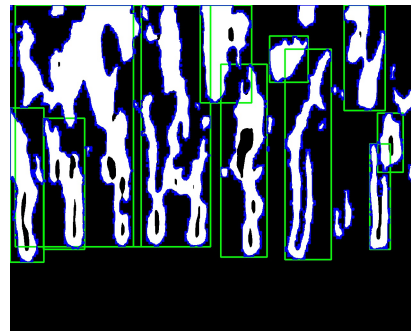


Figura 5.14: Contornos detectados después de realizar el filtrado por perímetro.

Los extremos de las imágenes no se tendrán en cuenta para el conteo, además de que en algunas ocasiones pueden dar lugar a resultados erróneos (podría haber manchas cercanas al

borde que fueran detectadas como capilares por su perímetro), por lo que se recortarán. El proceso de recorte es el siguiente: se obtienen las coordenadas de los contornos de interés y se selecciona el área que comprende desde la cabeza del capilar más externo hasta un 50 % del ancho de la imagen más hacia arriba. Así se encuadra la zona central, que es la que contiene la información que se va a procesar (Figura 5.15).

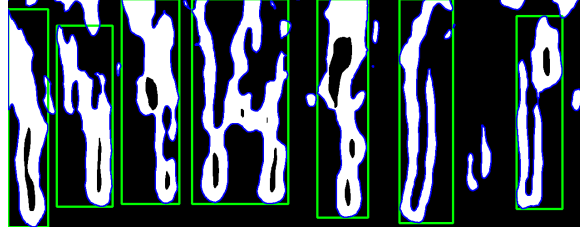


Figura 5.15: Imagen con los extremos recortados.

Aun así, hay casos en los que se detectan varios capilares que están unidos por alguna de sus partes como si fuese uno solo (en la Figura 5.13 se aprecia uno de esos casos). Para evitar esto, lo que se hace es seleccionar cada uno de los contornos obtenidos para seccionarlo y dejar tan sólo la cabeza del capilar, siguiendo un parámetro ρ para el porcentaje de recorte. De esta forma, si hubiese varios vasos en el mismo componente se verían por separado (Figura 5.16).

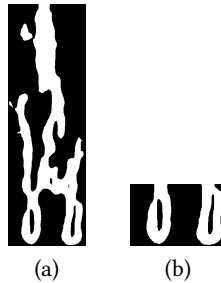


Figura 5.16: (a) Capilares pertenecientes a un solo contorno. (b) Contorno recortado para poder separar los capilares.

A continuación, se procede a contar de nuevo los componentes conexos que hay tras la separación. De esta forma se obtendrá el número total de vasos de cada imagen.

5.3.3 Resultados

Para la detección de la orientación, se ha establecido como valor óptimo $\kappa = 0.75$, de esta forma, se compara el 75 % superior de la imagen con el 25 % restante.

En el filtrado de ruido por perímetro, se ha observado que las estructuras de interés superan siempre un valor $v > 500$, por lo que se descartarán todos aquellos contornos que estén

por debajo de esta cifra. Se han hecho experimentos realizando el conteo con valores de cribado por perímetro diferentes. Con $v < 500$ no se eliminaban el ruido de la umbralización por completo y, por consiguiente, se detectaban muchos más capilares de los que realmente había. También se probó con valores mayores, pero esto hacía que se descartasen capilares en algunas imágenes.

Del mismo modo se varió el porcentaje de área que se recortaba para detectar varios capilares en un mismo contorno. Con un recorte de $\rho = 0.85$, en algunas ocasiones se recortaba por completo la cabeza de alguno de los capilares, por lo que no se tenía en cuenta. Con valores menores ($\rho = 0.75$) se incluía en la imagen la unión de los vasos, por lo que seguían detectándose como una sola estructura. Así, se estableció que el valor óptimo de recorte era de $\rho = 0.8$.

Se han comparado los resultados obtenidos en el sistema con los de un conteo manual de los capilares de todas las imágenes y el algoritmo coincidió un 90.47 % sobre un total de 336 capilares. Se obtuvieron, además, 14 Falsos Positivos. Generalmente el error es de ± 1 capilar por imagen. En la Figura 5.17 se puede apreciar la detección de los capilares (encuadrados en verde) en las imágenes mostradas. Aun así, el sistema aumenta el error en algunas excepciones que hasta para un usuario sería confuso el recuento (Figura 5.18), en las cuales el sistema falla en ± 2 capilares de media.

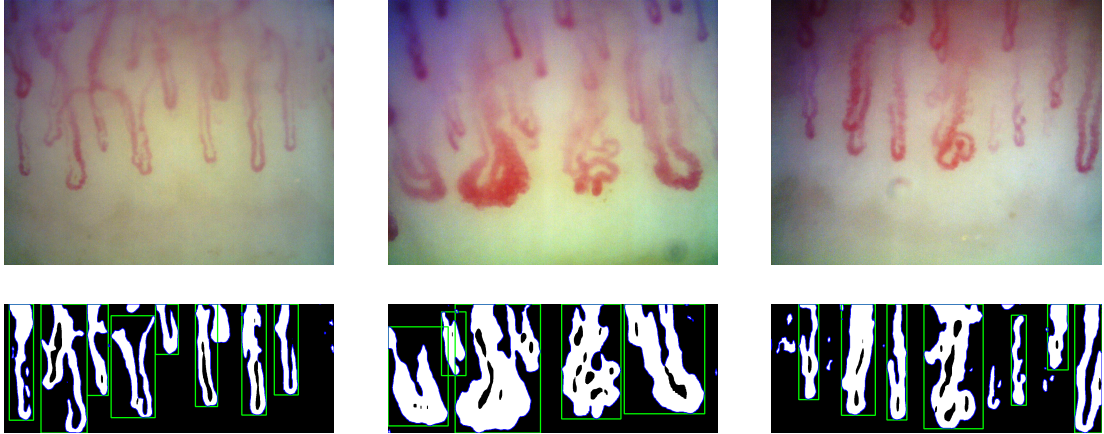


Figura 5.17: Ejemplo de detección de capilares. Fila superior: Imágenes originales. Fila inferior: Imágenes correspondientes segmentadas con los capilares detectados encuadrados.

5.4 Clasificación

En este paso se procede a estudiar la estructura de cada uno de los capilares detectados y en base a dicho estudio se clasificarán en *capilares sanos* y *capilares anómalos*. Para ello, se partirá de las estructuras conexas obtenidas en el proceso de conteo, tal y como se presentan

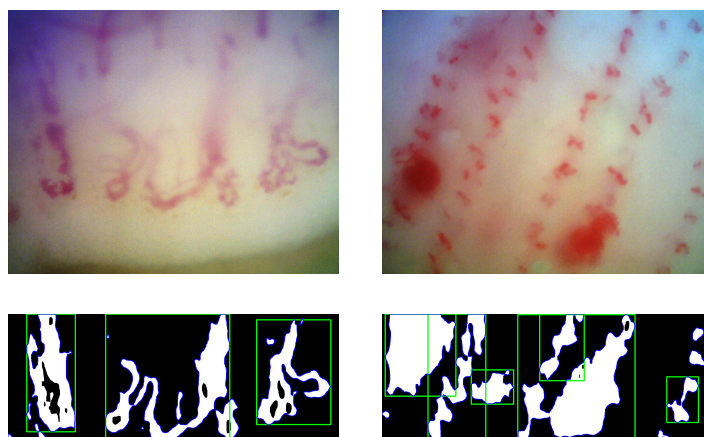


Figura 5.18: Ejemplo de detección fallida de capilares. Fila superior: Imágenes originales. Fila inferior: Imágenes correspondientes segmentadas con los capilares detectados encuadrados.

en la imagen (a) de la Figura 5.16.

Se ha mencionado con anterioridad que para realizar el diagnóstico, los expertos observan factores morfológicos tales como la longitud, la anchura y la regularidad de los vasos, por lo que esos tres parámetros serán los que se estudien en el algoritmo de clasificación. En las siguientes subsecciones se analizarán los procesos para obtener cada uno de ellos en detalle.

Para obtener el ancho y la regularidad, al igual que para el conteo, las estructuras se recortarán por la base, ya que no aporta información interesante. Sin embargo, el porcentaje de contorno que se recorte será diferente; en este caso será del 30% y no del 80%. Se sacrificarán algunas de las estructuras que contenían varios capilares unidos, puesto que, si bien capilares unidos serán detectados como uno solo, se podrán estudiar todo tipo de posibles tortuosidades que provocan su fusión.

Se han realizado dos aproximaciones para el cálculo de regularidad de los contornos: obteniendo la variabilidad de las direcciones del gradiente y calculando la relación entre el área del rectángulo mínimo que lo encuadra (*bounding box*) y el recuento de píxeles blancos en él.

5.4.1 Obtención de la longitud

Para realizar la medición de la longitud del capilar, en primer lugar, para cada contorno se calcula la *bounding box*. Luego, se calcula la longitud de dicha *bounding box* rotada para que sea perpendicular al eje x . Se establecerá un umbral λ para determinar si un capilar es anómalo o no (Figura 5.19).

Aun así, los capilares anómalos en algunos casos superan este umbral λ (en la Figura 5.20 se puede apreciar uno de ellos marcado en azul), por eso se ha decidido incluir más parámetros para clasificar correctamente cada vaso.

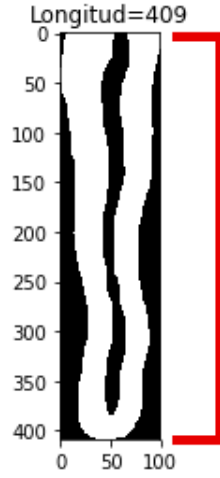


Figura 5.19: Cálculo de longitud del capilar.



Figura 5.20: Ejemplo de capilar anómalo largo (marcado en azul).

5.4.2 Obtención del ancho

Como se ha mencionado, se recortarán los contornos a la mitad para medir la parte de la cabeza del capilar (Figura 5.21). De esta forma, se tendrá un recorte más preciso de la zona de interés.

Partiendo del área recortada, se calculará su centroide o centro de masas. El centroide es la media aritmética de todos los puntos de una función. Si la función está compuesta de n puntos $x_1 \dots x_n$, entonces el cálculo del centroide sigue la siguiente ecuación:

$$centroide = \frac{1}{n} \sum_{i=1}^n x_i \quad (5.12)$$

Aunque en este caso, se aplica a estructuras unidimensionales. En el caso de las imágenes, las formas están compuestas por puntos (bidimensionales), por lo que la forma de hallar el centroide es un tanto diferente. En el procesamiento de imágenes, se hace uso de *momentos*

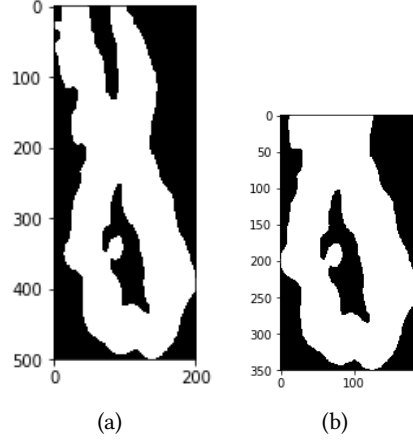


Figura 5.21: (a) Contorno entero. (b) Contorno recortado al 70 %.

de imagen [39]. Un momento de imagen es un número calculado según una fórmula de media ponderada con el que se pueden encontrar propiedades específicas de una imagen (radios, áreas, centroide...). La fórmula matemática para el cálculo del momento de orden n de un punto c es la siguiente:

$$M_n = \int_{-\infty}^{+\infty} (x - c)^n (f(x)) dx \quad (5.13)$$

siendo $f(x)$ la función de la que se desea conocer el centroide.

En caso de una imagen bidimensional son necesarias dos variables independientes:

$$M_{m,n} = \int \int (x - c_x)^m (y - c_y)^n f(x, y) dy dx \quad (5.14)$$

siendo $f(x, y)$ la imagen que se desea analizar.

Como las imágenes no están definidas en un dominio continuo, se puede discretizar el cálculo de los momentos:

$$M_{m,n} = \sum_{x=0}^{\infty} \sum_{y=0}^{\infty} (x - c_x)^m (y - c_y)^n f(x, y) \quad (5.15)$$

Para el cálculo del centroide se usarán los momentos $M_{1,0}$, $M_{0,0}$ y $M_{0,1}$ de la siguiente forma:

$$centroide_x = \frac{M_{1,0}}{M_{0,0}} \quad (5.16)$$

$$centroide_y = \frac{M_{0,1}}{M_{0,0}} \quad (5.17)$$

Estas dos operaciones darán la coordenada x y la coordenada y del centroide. En la Figura

5.22 se puede ver la localización del centroide de la imagen usada en la Figura 5.21 marcado con un punto rojo.

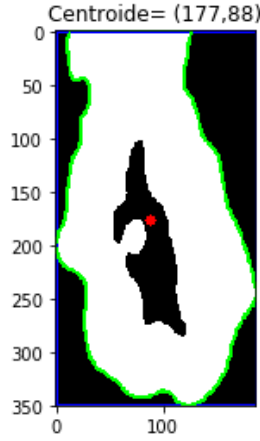


Figura 5.22: Centroide de la imagen.

A continuación, se contará el número de píxeles blancos que hay en la fila del centroide (Figura 5.23). Se utilizará un umbral ϕ para determinar si un capilar es anómalo o no.

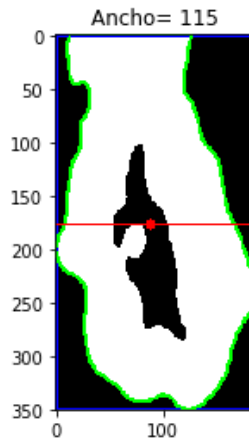


Figura 5.23: Recuento de píxeles blancos de la fila del centroide.

Esta es una métrica bastante fiable por sí sola, puesto que la mayoría de los capilares anómalos son anchos. Aun así, se ha decidido añadir el parámetro de regularidad, para reafirmar la clasificación.

5.4.3 Regularidad según variabilidad de orientaciones del gradiente

En esta primera aproximación para el cálculo de la regularidad del contorno del capilar se hace uso de la matriz de orientaciones del gradiente de la imagen. Para calcularla, primero

se filtra la imagen usando algún operador de cálculo de gradiente, como por ejemplo el operador de Prewitt, Sobel o Roberts. En nuestro caso se usa Sobel. Este operador se basa en la convolución de la imagen original con las matrices:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (5.18)$$

Convolucionando la imagen con G_x se obtendrá el gradiente en x y con G_y el gradiente en y .

Cuando se ha calculado el gradiente, se procede a calcular las orientaciones. Para ello, se sigue la siguiente fórmula:

$$\theta(x, y) = \arctan\left(\frac{I_y(x, y)}{I_x(x, y)}\right) \quad (5.19)$$

siendo (x, y) las coordenadas del píxel a calcular, I_x la imagen del gradiente en x y I_y la imagen del gradiente en y .

Entonces se comprueba cuál es la orientación predominante en la imagen y se calcula qué porcentaje de píxeles se desvían de ella más de τ grados:

$$r_a(x, y) = \begin{cases} 1 & \text{si } |\theta(x, y) - \theta_I| > \tau \\ 0 & \text{en caso contrario} \end{cases} \quad (5.20)$$

$$R_G = \sum_{x, y} r_a(x, y) \quad (5.21)$$

donde $r_a(x, y)$ es la desviación del píxel que se analiza, θ_I es la orientación predominante en la imagen y R_G el número total de píxeles que se desvían más de τ grados.

5.4.4 Regularidad según área de la *bounding box*

En este método, lo primero que se hace es calcular la *bounding box* del contorno recortado. Este cálculo da como resultado los cuatro puntos que encierran la figura, por lo que se hará uso de la fórmula de Green para el cálculo del área de polígonos:

$$A = \sum_{k=0}^n \frac{(x_{k+1} + x_k)(y_{k+1} + y_k)}{2} \quad (5.22)$$

donde n es el número de vértices, (x_k, y_k) es el punto número k y $(x_{n+1}, y_{n+1}) = (x_0, y_0)$, es decir, que el primer vértice se encuentra tanto al principio como al final de la lista de vértices.

Luego, se cuentan los píxeles blancos que hay dentro de esa área de la imagen y se dividen

por el total:

$$R_{BB} = \frac{\sum_{x,y} I(x,y)}{A} \quad (5.23)$$

Así se obtiene un porcentaje del espacio ocupado por el contorno. En los contornos más regulares como el de la Figura 5.19, el porcentaje será elevado, mientras que en los que presentan tortuosidades, como el de la Figura 5.24 será inferior.

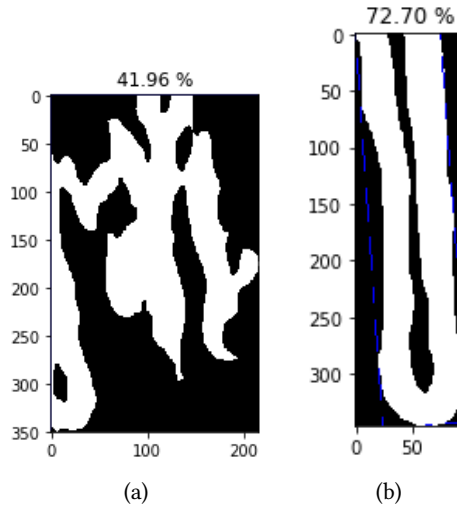


Figura 5.24: (a) Porcentaje obtenido en capilar irregular (41.96 %). (b) Porcentaje obtenido en capilar regular (72.70 %).

5.4.5 Resultados

Se ha descartado el uso de un clasificador entrenado para la realización de esta tarea, ya que la base de datos de la que se dispone no está bien balanceada (sólo hay aproximadamente un 25% de capilares sanos) y no cuenta con un número suficiente de ejemplos para una división en conjuntos de entrenamiento, validación y test que pudiera dar buenos resultados. Además se trata de un problema sencillo, ya que los capilares anómalos son claramente distinguibles de los sanos basándonos en las métricas definidas, por este motivo se utilizarán umbrales.

Se ha analizado la longitud de los capilares y se ha observado que aquellos que se pueden considerar pequeños a simple vista no superan nunca $\lambda = 250$ píxeles, por lo que se considerará este umbral para determinar si un vaso es largo o no. Se han realizado experimentos estableciendo $\lambda = 300$ píxeles, pero se descartó porque el recorte efectuado a los contornos provocaba que algunos capilares sanos no alcanzasen este umbral. La diferencia de longitud se puede apreciar en la Figura 5.25. En este caso, el capilar rodeado más pequeño, tiene 129 píxeles de longitud y el más largo, 306.



Figura 5.25: Diferencias de longitudes en los capilares. Capilar rodeado más pequeño: 129 píxeles de longitud. Capilar rodeado más largo: 306 píxeles de longitud.

Del mismo modo, se ha observado la anchura de los capilares y aquellos que se consideran finos a simple vista nunca superan el umbral $\phi = 70$ píxeles, por eso se considerará un capilar ancho aquel que supere este número. Se han hecho pruebas variando los valores de ϕ , estableciendo el umbral de anchura en $\phi = 60$ píxeles, pero en algunas ocasiones se tomaban capilares aparentemente sanos como si fuesen anómalos. También se ha aumentado a $\phi = 80$ píxeles, pero el porcentaje de acierto no superaba el actual.

En el caso del cálculo de la regularidad basada en orientaciones del gradiente, se estableció un $\tau = 20$ grados. Esta aproximación no dio buenos resultados por sí sola, puesto que si un capilar resultaba ser muy tortuoso, pero las tortuosidades tenían la misma orientación que el propio vaso, su regularidad era alta (Figura 5.26). De ahí que se ideara otra forma de calcularla.

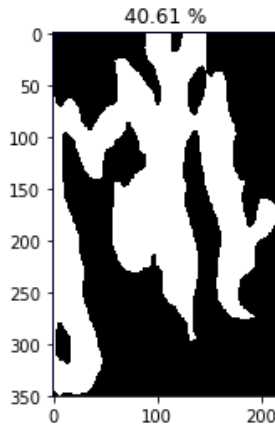


Figura 5.26: Capilar tortuoso con una regularidad bastante alta (40.61 %) usando el método de cálculo de orientaciones del gradiente.

Para el cálculo de regularidad basada en el área de la *bounding box*, se estableció como valor determinante un 50 % de área rellena.

El porcentaje de regularidad final se ha calculado ponderando los dos métodos anterior-

mente descritos de la siguiente forma:

$$R = 0.3R_G + 0.7R_{BB} \quad (5.24)$$

Para que el capilar se considere regular será preciso que alcance al menos un $R = 0.55$. Se han considerado ambas aproximaciones, ya que el cálculo de orientaciones de gradiente podría ayudar a reafirmar la irregularidad de un capilar anómalo que, debido a su grosor, supere el umbral de área de la *bounding box* (Figura 5.27).

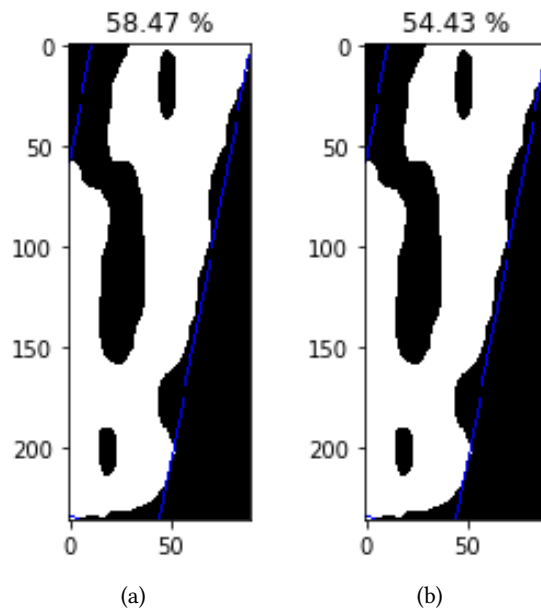


Figura 5.27: (a) Porcentaje de regularidad obtenido con el método de la *bounding box* (58.47 %). (b) Porcentaje de regularidad obtenido con el método mixto (54.45 %).

Los criterios utilizados para cada una de las métricas se pueden ver resumidos en la Tabla 5.10:

Tabla 5.10: Valores para determinar la clasificación.

Métrica	Criterio utilizado
Capilar ancho	$\phi > 70$
Capilar largo	$\lambda > 250$
Capilar regular	$R > 0.55$

Si los capilares cumplen los tres criterios, serán clasificados como *capilares sanos*.

Se han comparado los resultados obtenidos en el sistema con los de un diagnóstico manual de los capilares de todas las imágenes y el algoritmo coincidió en un 90.73 % de los casos. En la Figura 5.28 se pueden observar tres casos en los que el algoritmo falla y en la Figura 5.29, tres

en los que acierta. Generalmente, las clasificaciones incorrectas se deben a problemas en la localización del centroide o el cálculo de regularidad. En los casos de la Figura 5.28, éstas son las causas de los errores: en el primer capilar, el centroide se encuentra justo en el agujero que hay en medio de la imagen y eso hace que el ancho sea aceptable (53 píxeles). En el segundo, el centroide se posiciona en un área que es más ancha que la media del vaso, justo por debajo del agujero que hay en la parte derecha de la imagen, dando un ancho no válido de 71 píxeles. En el último caso, la clasificación incorrecta se debe al cálculo de la regularidad. La base del capilar es más ancha que el resto del cuerpo, lo que hace que la *bounding box* sea extensa. Por esto no se alcanza un porcentaje de relleno aceptable (52 %).

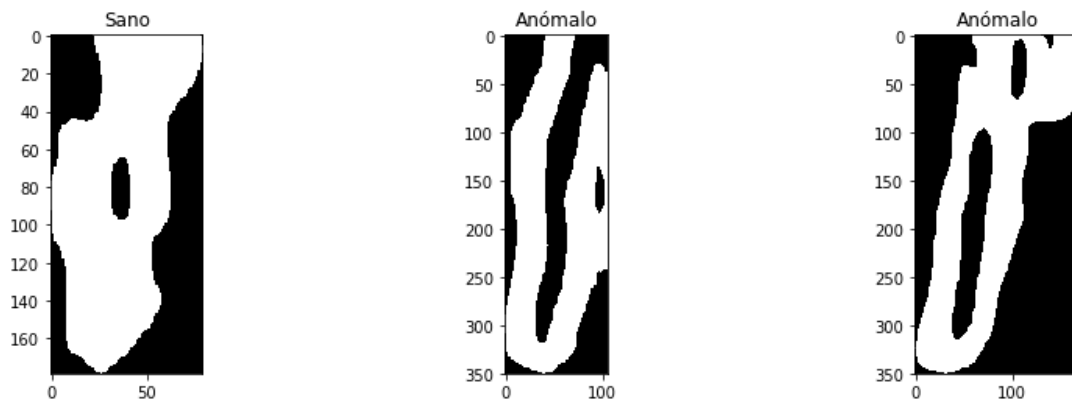


Figura 5.28: Ejemplo de clasificación errónea de capilares.

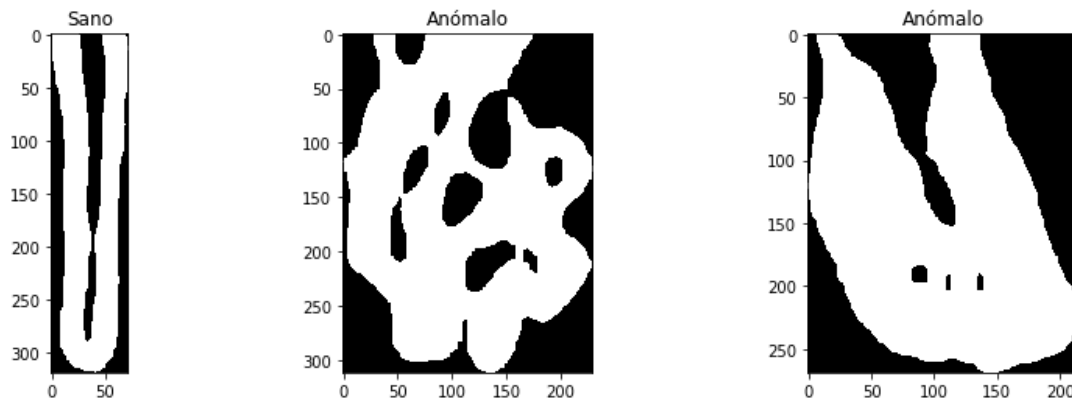


Figura 5.29: Ejemplo de clasificación correcta de capilares.

Aplicación web

En este capítulo se verá la estructura de la aplicación web que servirá de interfaz para la metodología anteriormente descrita. Se detallará su análisis, diseño, implementación y las pruebas realizadas.

6.1 Análisis

Se han analizado los requisitos funcionales que tendrá la aplicación y se han identificado dos actores: el usuario anónimo y el usuario registrado. Al primero tan solo se le permitirá registrarse en la aplicación, mientras que el segundo podrá realizar las siguientes acciones:

- Identificarse en la aplicación.
- Ver la lista de pacientes.
- Registrar pacientes nuevos.
- Ver el perfil de los pacientes.
- Modificar el perfil de los pacientes.
- Buscar pacientes por nombre y apellidos.
- Ver su propio perfil.
- Modificar su propio perfil.
- Subir capilaroscopias de los pacientes para ver su diagnóstico.
- Ver los diagnósticos de cada capilaroscopia.

En la Figura 6.1 se puede ver el diagrama de casos de uso y en las Tablas de la 6.1 a la 6.11 se describirán en detalle cada uno de ellos.

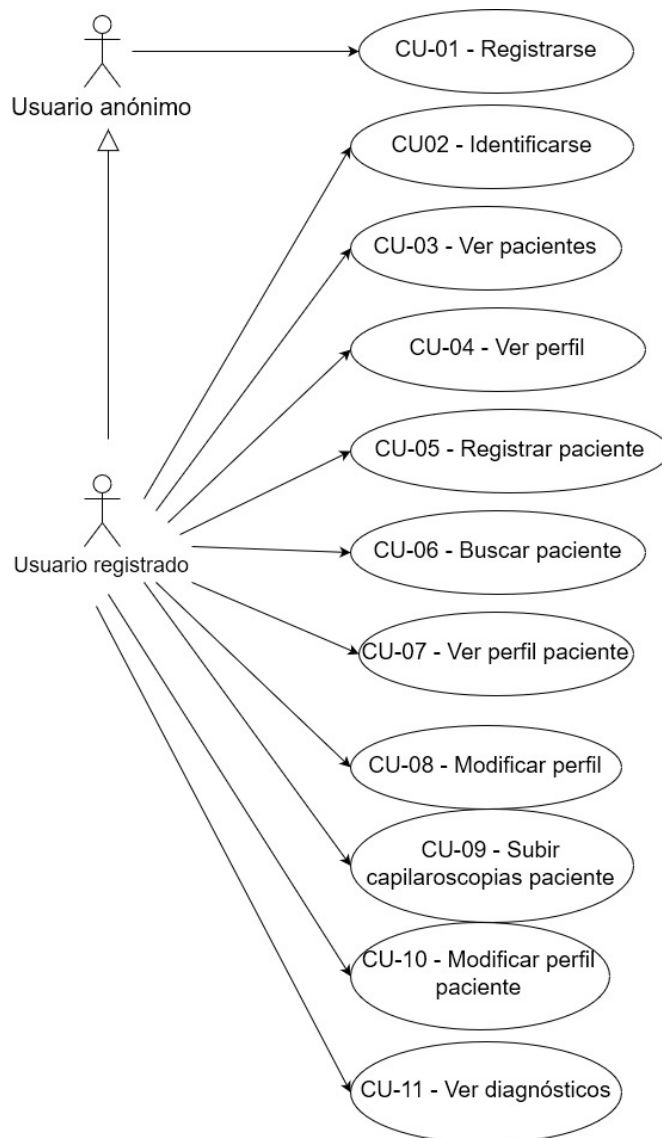


Figura 6.1: Casos de uso detectados para este proyecto.

Tabla 6.1: CU-01: Registrarse.

CU-01	Registrarse
Objetivos asociados	Registrarse en el sistema.
Descripción	El sistema registra los datos introducidos añadiendo un nuevo usuario al sistema.
Precondición	-
Secuencia normal	<ol style="list-style-type: none">1. El sistema solicita al usuario los siguientes campos:<ul style="list-style-type: none">• Nombre.• Apellidos.• DNI.• E-mail.• Contraseña.• Repetir contraseña.2. El usuario cubre los campos y pulsa registrar.3. El sistema verifica los datos y si son correctos registra al usuario en el sistema, volviendo a la pantalla de login.
Postcondición	El usuario queda registrado y este caso de uso no podrá volver a ser lanzado por el mismo usuario.
Excepciones	Paso 3: Si ya existe un usuario con el correo electrónico introducido, se muestra un error y se vuelve al paso 1. Paso 3: Si ya existe un usuario con el DNI introducido, se muestra un error y se vuelve al paso 1.
Importancia	Alta

Tabla 6.2: CU-02: Identificarse.

CU-02	Identificarse
Objetivos asociados	Identificarse en el sistema.
Descripción	El usuario se identifica en el sistema.
Precondición	-
Secuencia normal	<ol style="list-style-type: none">1. El sistema solicita al usuario los siguientes campos:<ul style="list-style-type: none">• E-mail.• Contraseña.2. El usuario cubre los campos y pulsa “Iniciar sesión”.3. El sistema verifica los datos.
Postcondición	El usuario queda logueado.
Excepciones	Paso 3: Si los datos no son correctos se vuelve al paso 1.
Importancia	Alta

Tabla 6.3: CU-03: Ver pacientes.

CU-03	Ver pacientes
Objetivos asociados	Ver la lista de los pacientes.
Descripción	Se muestra una lista con los detalles de todos los pacientes del sistema, que podrán ser ordenados por varios criterios.
Precondición	El usuario debe estar identificado
Secuencia normal	<ol style="list-style-type: none"> 1. El sistema muestra al usuario la lista de los pacientes que hay guardados en el sistema. Dicha lista podrá ser ordenada por : <ul style="list-style-type: none"> • Nombre. • Apellidos (por defecto). • El médico que lo ha registrado. • Fecha de ingreso en el sistema.
Postcondición	Se muestran los datos de los pacientes.
Excepciones	-
Importancia	Media

Tabla 6.4: CU-04: Ver perfil.

CU-04	Ver perfil
Objetivos asociados	Ver el perfil del usuario.
Descripción	Se muestra la información que usuario ha guardado en el sistema.
Precondición	El usuario debe estar identificado
Secuencia normal	<ol style="list-style-type: none"> 1. El sistema muestra al usuario los siguientes datos: <ul style="list-style-type: none"> • Nombre. • Apellidos. • E-mail.
Postcondición	Se muestran los datos del usuario.
Excepciones	-
Importancia	Baja

Tabla 6.5: CU-05: Registrar paciente.

CU-05	Registrar paciente
Objetivos asociados	Registrar un nuevo paciente en el sistema.
Descripción	El sistema registra los datos introducidos añadiendo un nuevo paciente al sistema.
Precondición	El usuario debe estar identificado
Secuencia normal	<ol style="list-style-type: none"> 1. El sistema solicita al usuario los siguientes campos: <ul style="list-style-type: none"> • Nombre. • Apellidos. • DNI. • Teléfono. • E-mail. • Antecedentes médicos. 2. El usuario cubre los campos y pulsa registrar. 3. El sistema verifica los datos y si son correctos registra al paciente en el sistema, volviendo a la pantalla de buscar pacientes.
Postcondición	El paciente queda registrado y este caso de uso no podrá volver a ser lanzado para el mismo paciente.
Excepciones	Paso 3: Si ya existe un paciente con el DNI introducido, se muestra un error y se vuelve al paso 1.
Importancia	Alta

Tabla 6.6: CU-06: Buscar paciente.

CU-06	Buscar paciente
Objetivos asociados	Buscar paciente.
Descripción	El sistema busca al paciente que se le indica, bien por nombre, por apellidos o por el nombre del médico que lo ha registrado.
Precondición	El usuario debe estar identificado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario establece los criterios de búsqueda. 2. El sistema busca al paciente correspondiente con los datos introducidos por el usuario. 3. El sistema: <ul style="list-style-type: none"> • Si encuentra el paciente, muestra sus datos. • Si no encuentra el paciente, enseña una lista vacía.
Postcondición	-
Excepciones	-
Importancia	Alta

Tabla 6.7: CU-07: Ver perfil paciente.

CU-07	Ver perfil paciente
Objetivos asociados	Ver el perfil del paciente.
Descripción	Se muestra la información que hay guardada en el sistema acerca del paciente seleccionado.
Precondición	El usuario debe estar identificado y ha seleccionado un paciente.
Secuencia normal	<ol style="list-style-type: none"> 1. El sistema muestra al usuario los siguientes datos del paciente: <ul style="list-style-type: none"> • Nombre. • Apellidos. • E-mail. • Teléfono. • Médico que lo ha registrado. • Antecedentes médicos. • Una lista con las capilaroscopias que se han guardado de dicho paciente.
Postcondición	Se muestran los datos del paciente.
Excepciones	-
Importancia	Alta

Tabla 6.8: CU-08: Modificar perfil.

CU-08	Modificar perfil
Objetivos asociados	Modificar el perfil del usuario.
Descripción	Se muestra la información del usuario que hay guardada en el sistema y se le permite modificarla.
Precondición	El usuario debe estar identificado.
Secuencia normal	<ol style="list-style-type: none"> 1. El sistema muestra al usuario los siguientes datos, que podrá modificar: <ul style="list-style-type: none"> • Nombre. • Apellidos. • E-mail. • Contraseña 2. El usuario envía los datos modificados. 3. El sistema verifica los datos y si son correctos guarda los cambios.
Postcondición	Se guardan las modificaciones de los datos del usuario.
Excepciones	Paso 3: Si ya existe un usuario con el correo electrónico introducido, se muestra un error y se vuelve al paso 1.
Importancia	Baja

Tabla 6.9: CU-09: Subir capilaroscopias paciente.

CU-09	Subir capilaroscopias paciente
Objetivos asociados	Subir capilaroscopias del paciente.
Descripción	Se podrán añadir capilaroscopias del paciente seleccionado al sistema para su diagnóstico.
Precondición	El usuario debe estar identificado y ha seleccionado un paciente.
Secuencia normal	<ol style="list-style-type: none"> 1. El sistema solicita al usuario las capilaroscopias que se desean añadir. 2. El sistema realiza el diagnóstico. 3. El sistema guarda tanto la capilaroscopia como el diagnóstico.
Postcondición	Se añaden la capilaroscopia y el diagnóstico al sistema.
Excepciones	-
Importancia	Alta

Tabla 6.10: CU-10: Modificar perfil paciente.

CU-10	Modificar perfil paciente
Objetivos asociados	Modificar el perfil del paciente.
Descripción	Se muestra la información del paciente que hay guardada en el sistema y se le permite modificarla.
Precondición	El usuario debe estar identificado y debe haber seleccionado un paciente.
Secuencia normal	<ol style="list-style-type: none"> 1. El sistema muestra al usuario los siguientes datos del paciente, que podrá modificar: <ul style="list-style-type: none"> • Nombre. • Apellidos. • Teléfono • E-mail. • Antecedentes médicos 2. El usuario envía los datos modificados. 3. El sistema verifica los datos y si son correctos guarda los cambios.
Postcondición	Se guardan las modificaciones de los datos del usuario.
Excepciones	-
Importancia	Baja

Tabla 6.11: CU-11: Ver diagnósticos paciente.

CU-11	Ver diagnósticos
Objetivos asociados	Ver diagnósticos de las capilaroscopias.
Descripción	Se muestra, para la capilaroscopia seleccionada, los vasos que se han detectado y el diagnóstico de cada uno.
Precondición	El usuario debe estar identificado y ha seleccionado un paciente y una capilaroscopia.
Secuencia normal	<ol style="list-style-type: none"> 1. El sistema muestra al usuario los siguientes datos de la capilaroscopia: <ul style="list-style-type: none"> • Número de capilares. • Imagen de cada capilar. • Diagnóstico de cada capilar.
Postcondición	Se muestran los datos de la capilaroscopia.
Excepciones	-
Importancia	Alta

6.2 Diseño

En este apartado, se detallará el diseño propuesto para el desarrollo de la aplicación web. Se incluirán las maquetas realizadas para la interfaz gráfica, una descripción de los patrones de diseño utilizados y el diagrama de la base de datos.

6.2.1 Maquetación

A partir de los requisitos detectados en el apartado de análisis, se han elaborado una serie de pantallas para la interfaz gráfica de la web. En la Figura 6.2 se puede ver la pantalla de inicio de sesión. Dispone de un botón para ir al formulario de registro para un nuevo usuario.

Figura 6.2: Pantalla de inicio de sesión.

La Figura 6.3 muestra la pantalla de registro para un nuevo usuario. Todos los campos que se piden son obligatorios.

En la Figura 6.4 se muestra la pantalla de inicio, en la que se puede ver la lista de los pacientes. También se ha establecido una barra para la búsqueda de pacientes según el criterio seleccionado en la lista situada al lado, así como un botón para el registro de un nuevo paciente.

Se tendrá acceso al enlace que redirige a esta pantalla desde todas las páginas de la web. También será accesible desde todas partes el enlace al perfil del usuario y el enlace para cerrar sesión.

La Figura 6.5 se corresponde a la pantalla de registro de un nuevo paciente. Todos los campos serán obligatorios a excepción de los antecedentes médicos.

En la Figura 6.6 se puede observar la pantalla de perfil de un paciente. En la parte de abajo se sitúa un apartado donde aparecen las capilaroscopias hechas a ese paciente. Cada

LOGO

Nuevo Usuario

[<< Volver a Inicio de Sesión](#)

Datos personales

Nombre*:
Apellidos*:
DNI*:

Datos de inicio de sesión

E-mail*:

Contraseña*:

Repetir contraseña*:

Registrarse

Figura 6.3: Pantalla de registro de un usuario nuevo.

LOGO

Martín Sánchez

[Inicio](#)
[Mi perfil](#)
[Cerrar sesión](#)

Pacientes

Buscar por:

Apellidos

▼

Search

🔍

Registrar nuevo paciente






Nombre	Apellidos	Médico	Fecha de registro
 John	Rambo	Marcos Pérez	15 Sep, 8:56 AM (2013)
 Klaus	Meine	Martín Sánchez	15 Sep, 7:12 AM (2013)
 Pablo	Ruiz Picasso	Marcos Pérez	15 Sep, 4:32 AM (2013)
Jennifer	Pinsker	Marcos Pérez	15 Sep, 2:08 AM (2013)
 Art	Garfunkel	Silvia Martínez	15 Sep, 8:56 AM (2013)
 Alan	Parsons	Silvia Martínez	15 Sep, 7:12 AM (2013)

Figura 6.4: Pantalla de ver, buscar y registrar pacientes.

LOGO

Martín Sánchez

[Inicio](#) [Mi perfil](#) [Cerrar sesión](#)

Nuevo Paciente

Datos personales

Nombre*:

Apellidos*:

DNI*:

E-mail*:

Teléfono*:

Antecedentes médicos:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Registrar paciente

Figura 6.5: Pantalla de registro de un paciente nuevo.

imagen es un enlace que llevará a la pantalla de diagnóstico correspondiente. También se muestra un botón para añadir una nueva capilaroscopia que, al ser pulsado, pedirá al usuario el archivo que desea subir. En la esquina superior derecha se sitúa un botón de *Editar perfil* que si se pulsa, permitirá al usuario modificar los datos del paciente (o al menos aquellos que se puedan modificar).

The mockup shows a web interface for a patient profile. At the top, there's a header bar with a 'LOGO' placeholder on the left, the user's name 'Martín Sánchez' in the center, and navigation links 'Inicio', 'Mi perfil', and 'Cerrar sesión' on the right. Below the header, the main content area is titled 'Alan Parsons' with an edit icon (pencil) on the right. The content is divided into two main sections. The first section, 'Datos personales', contains a grid of labels and values: 'Nombre: Alan Parsons', 'E-mail: ammonia@avenue.com', 'Teléfono: 6XX XXX XXX', and 'Médico: Martín Sánchez'. To the right of these is a text area for 'Antecedentes médicos' containing placeholder text. The second section, 'Capilaroscopias', features a file upload icon (blue square with a plus) on the left, followed by a row of six placeholder boxes (squares with an 'X'). A vertical scrollbar is on the right side of this section.

Figura 6.6: Pantalla de perfil de un paciente.

La Figura 6.7 ilustra la pantalla que muestra los diagnósticos para los capilares de la capilaroscopia seleccionada, así como el número de vasos que se han detectado.

En la Figura 6.8 se puede observar la pantalla de perfil de usuario. El procedimiento de modificación de los datos es exactamente el mismo que el de la pantalla de perfil de paciente.

6.2.2 Patrones de diseño

A continuación se enumerarán los patrones que se han utilizado para el diseño de la aplicación web.

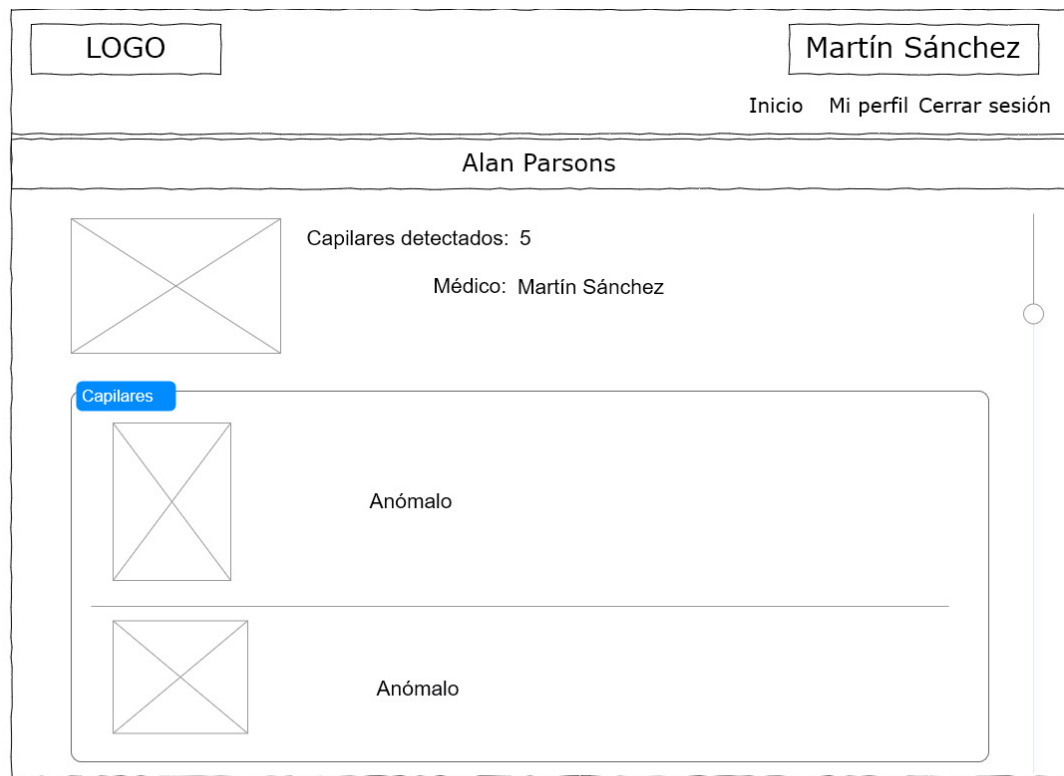


Figura 6.7: Pantalla de diagnóstico de una capilaroscopia.

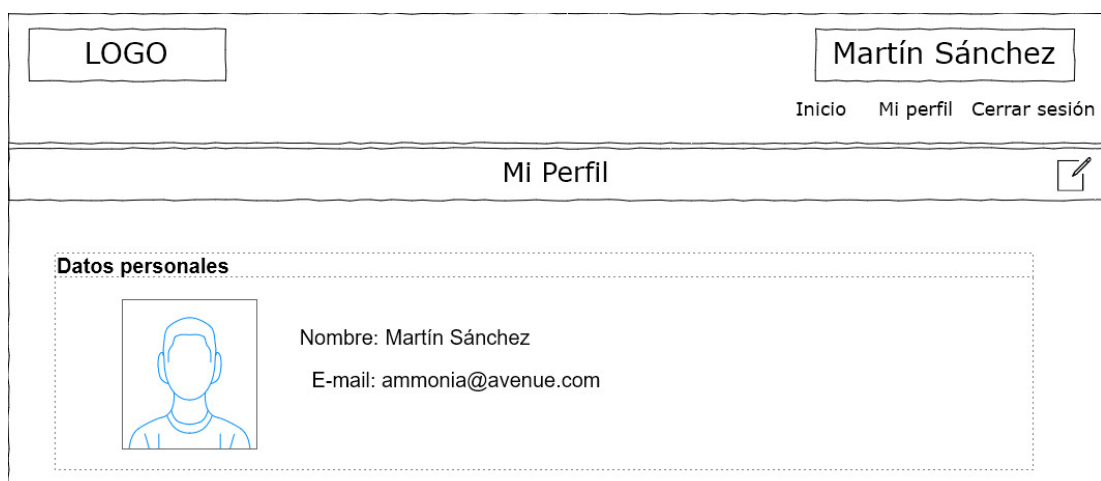


Figura 6.8: Pantalla de perfil de usuario.

Modelo-Vista-Controlador (MVC)

Modelo-Vista-Controlador es un patrón de arquitectura de software que, utilizando tres componentes (Modelos, Vistas y Controladores, como su nombre indica) separa la lógica de la aplicación de la lógica de la vista [40]. Es altamente usado en el diseño de aplicaciones gráficas. A continuación se comentará el funcionamiento de cada uno de los componentes:

- El Modelo: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones.
- La Vista: Presenta el modelo en un formato adecuado para la interacción (la interfaz de usuario), por tanto requiere de dicho modelo la información que debe representar como salida.
- El Controlador: Responde a eventos e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información (por ejemplo, editar un registro en una base de datos). También puede enviar comandos a su vista asociada si se solicita un cambio en la forma en que se presenta el modelo, por tanto se podría decir que el controlador hace de intermediario entre la vista y el modelo.

En la Figura 6.9 se plasma un esquema de su funcionamiento.

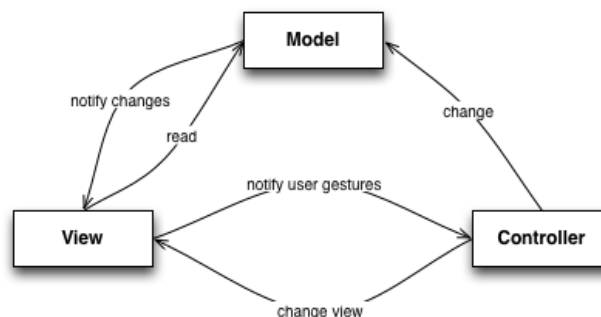


Figura 6.9: Esquema de Modelo-Vista-Controlador.

Patrón Instancia única

El patrón Instancia única (Singleton) se utiliza para asegurar que en una aplicación sólo habrá una instancia de una determinada clase y provee un punto de acceso global a ella. En la Figura 6.10 se muestra un esquema de su arquitectura [41].

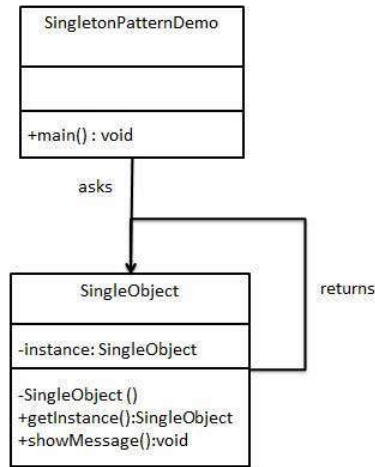


Figura 6.10: Esquema de patrón Instancia única.

6.2.3 Base de Datos

La aplicación contará con una base de datos que guardará la información de los usuarios y los pacientes, incluyendo las capilaroscopias y el diagnóstico de cada uno de los capilares detectados. De los usuarios se guardará el nombre, los apellidos, el DNI, email y una clave. De los pacientes se guardan nombre, apellidos, teléfono de contacto, email, antecedentes médicos, la fecha en la que se registró en el sistema y también el médico que lo registró. De las capilaroscopias se guarda el paciente al que pertenece, la imagen en sí, la fecha en la que se añadió al sistema, el médico que la añadió, si está diagnosticada o no y el número de capilares que se detectaron. Para cada capilar se guardará su imagen y su diagnóstico asociado. En la Figura 6.11 se puede observar el diagrama Entidad-Relación.

6.2.4 Arquitectura

A continuación se detallará la arquitectura de la aplicación web y la inclusión de los patrones. Para comenzar, se presentan las diferentes clases en la Figura 6.12. En este caso, la gestión de la vista la proporcionará el navegador web. Será el encargado de mostrar al usuario todos los elementos visuales. En el caso del controlador y el modelo, será la propia librería Django la que lleve a cabo la gestión. Django cuenta con tres módulos específicos para este propósito:

- **models.py**, que es donde se establecen las clases que se crearán en la base de datos (en este caso Usuario, Paciente, Capilaroscopia y Capilar). Django almacena las claves foráneas como objetos, de esta forma se proporciona una mayor facilidad y rapidez de acceso a sus atributos.
- **views.py**, que contiene la gestión de los datos del modelo para enviárselas luego a las

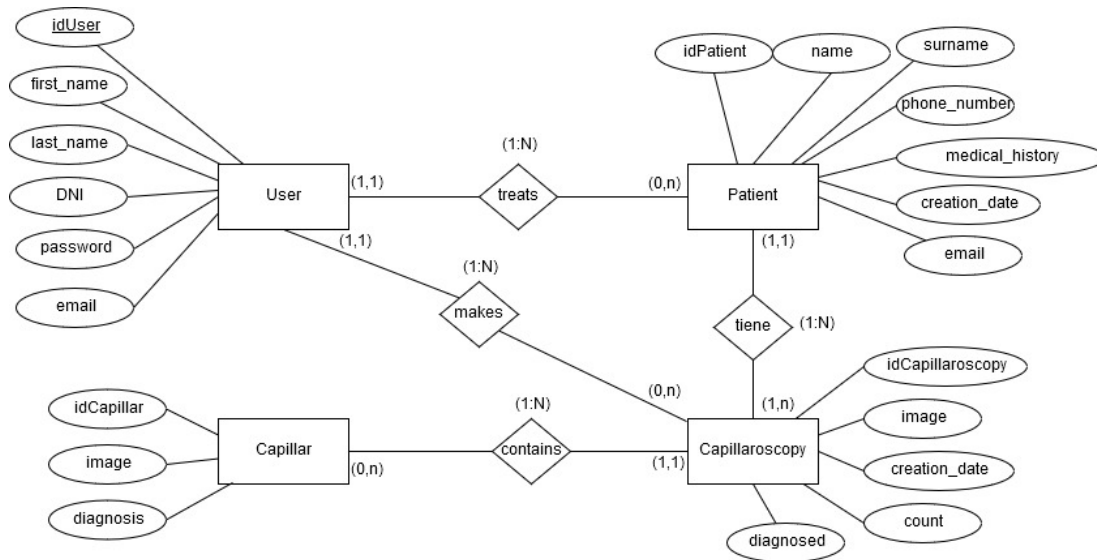


Figura 6.11: Diagrama Entidad-Relación.

diferentes vistas de la aplicación. La gestión de cada vista puede efectuarse creando una clase o con métodos específicos (se mostrará la estructura de este módulo más adelante).

- **forms.py**, que es donde se incluyen las definiciones de los formularios de los que hará uso la aplicación. En este caso se han creado formularios para el registro y edición de usuarios, el registro de pacientes y la subida de imágenes.

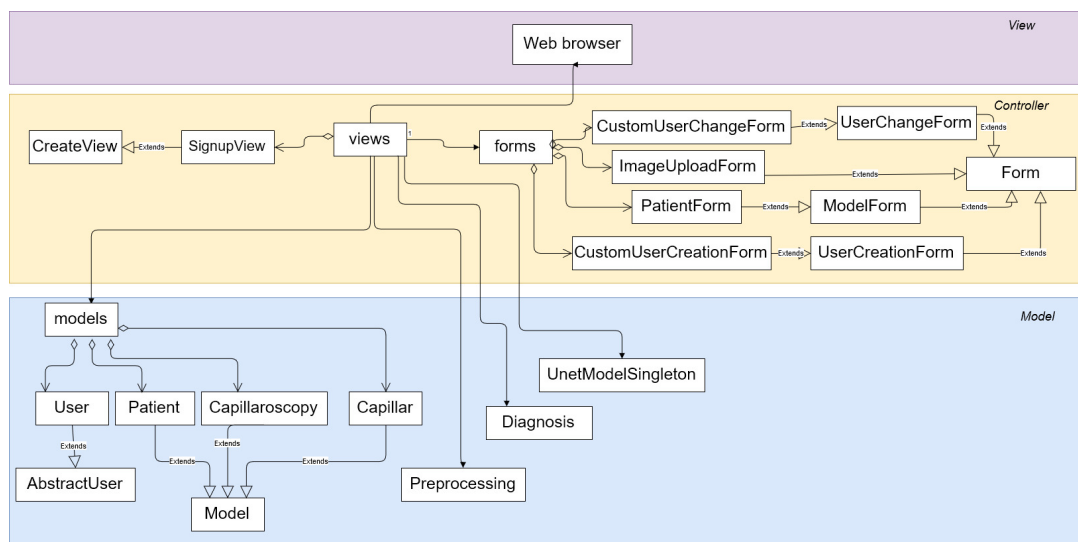


Figura 6.12: Diagrama de clases de la aplicación.

Como el mayor tiempo de procesado se encuentra en el diagnóstico de las capilaroscopias

(aproximadamente 6 segundos), y más concretamente en la carga del modelo de la U-Net en memoria, se ha establecido un patrón Instancia única para que solo se cargue una vez, al iniciar el servidor. Se ha creado una clase **UnetModelSingleton**, que cargará el modelo de la U-Net. De esta forma se reduce el tiempo de diagnóstico por imagen a 3 segundos. En la Figura 6.13 se puede observar el diseño de la clase **UnetModelSingleton**.

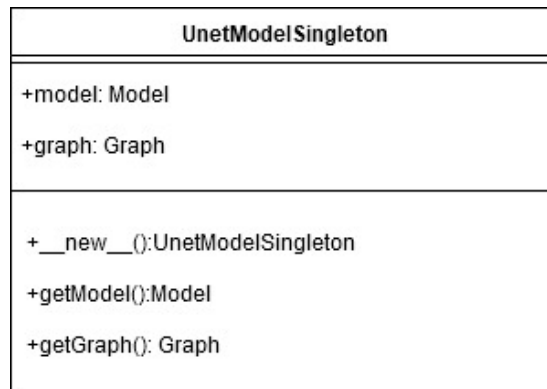


Figura 6.13: Diagrama de la clase **UnetModelSingleton**.

A continuación se adjunta un diagrama de secuencia que ilustra el proceso para diagnosticar una capilaroscopia (Figura 6.14). La secuencia se inicia cuando el usuario selecciona la imagen que desea diagnosticar. El navegador web enviará una petición al controlador, que preprocesará la imagen y luego hará uso del algoritmo de clasificación. Éste devolverá el conteo de los capilares detectados, así como sus formas y respectivos diagnósticos. En ese momento, el controlador hará una llamada al modelo para añadir cada uno de los capilares y su información a la base de datos. Una vez hecho esto, se modificará la información de la capilaroscopia seleccionada para añadir el conteo y se establecerá como diagnosticada. Al terminar, el controlador dará orden al navegador para que muestre la pantalla de diagnóstico.

6.3 Implementación

En este apartado se detallará el funcionamiento de la aplicación tras su diseño e implementación. Como se ha mencionado, la estructura del módulo **views** puede estar implementada de dos formas: mediante clases y con métodos. En este proyecto se han mezclado ambas, aprovechando la gestión automática que Django proporciona para ciertas vistas (por ejemplo, en la clase `SignupView`, que lleva a cabo el registro de usuarios). En los casos que requieren una manipulación de los datos antes de ser guardados y/o mostrados, y para los que la librería no proporciona una implementación predefinida, se han codificado funciones específicas. En la figura 6.15 se muestra el diagrama de este módulo (excluyendo la clase `SignupView`).

Se puede observar que todos los métodos reciben un objeto **HttpRequest**, que será donde

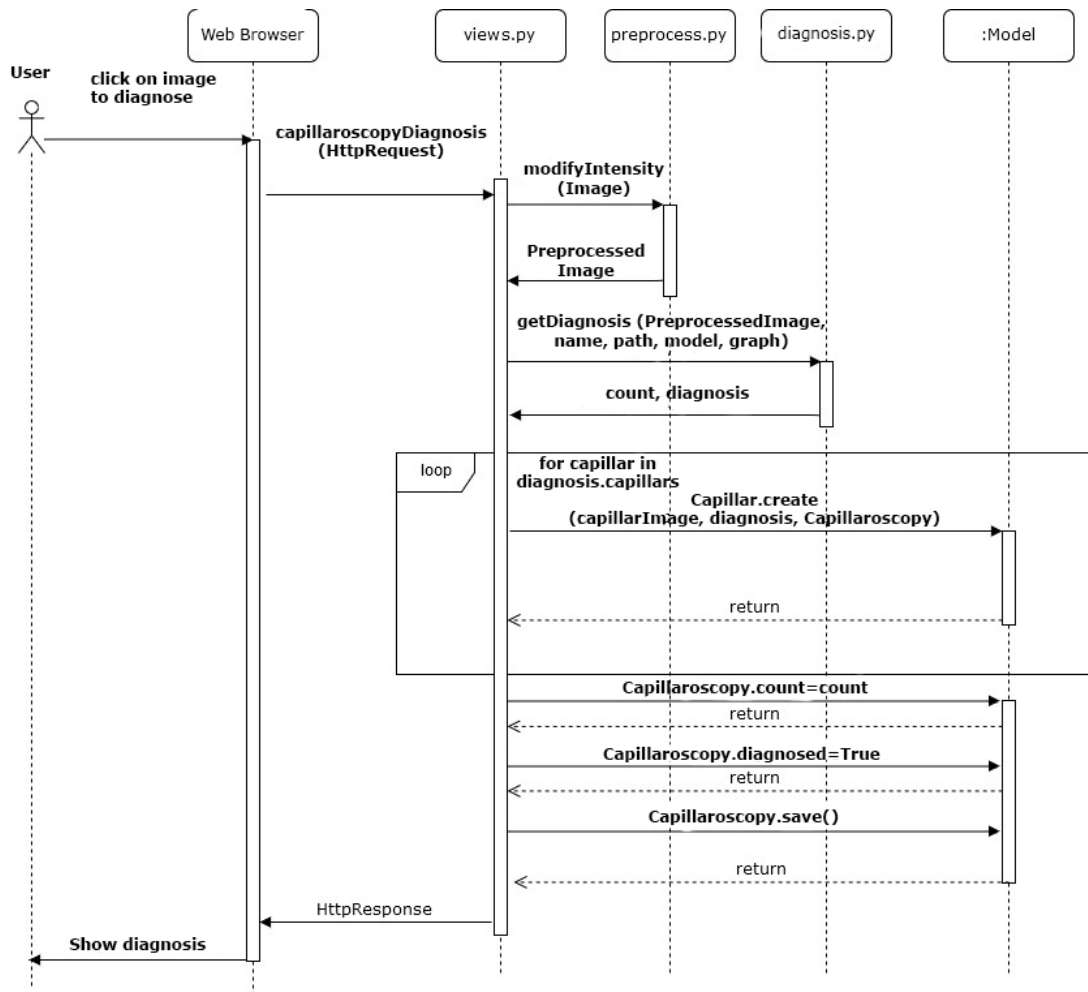
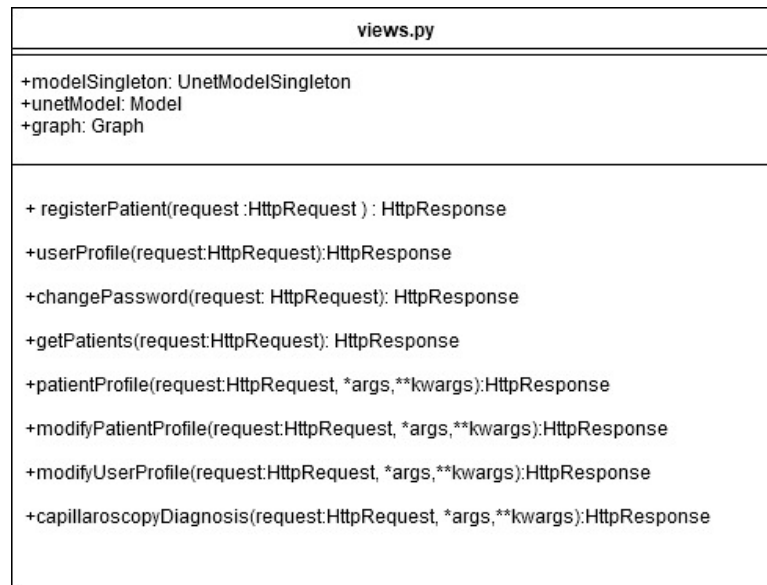


Figura 6.14: Diagrama de secuencia para diagnosticar una imagen.

se incluyan los datos que el usuario envíe a través del navegador, y todas devuelven un objeto **HttpResponse**, donde se adjuntarán los datos tras procesarlos para que el navegador los muestre.

Al entrar en la aplicación, el usuario verá la pantalla de inicio de sesión (Figura 6.16), en la que se le pedirá un email y una contraseña, o desde la que podrá entrar a la pantalla de registro de un nuevo usuario. Si se introducen datos erróneos, la aplicación mostrará un mensaje de error y vaciará el campo de contraseña. En la pantalla de registro (Figura 6.17) se pedirán los campos que el usuario debe cubrir para crear una cuenta en la aplicación. Todos los campos son obligatorios. El sistema comprobará que no se introduzca un email o un DNI que ya exista en la base de datos. También se asegurará de que la contraseña elegida cumpla ciertos requisitos: no podrá ser parecida a los datos personales, no podrá tener menos de ocho caracteres y no podrá estar constituida únicamente por números. Si todos los datos

Figura 6.15: Diagrama del módulo **views.py**.

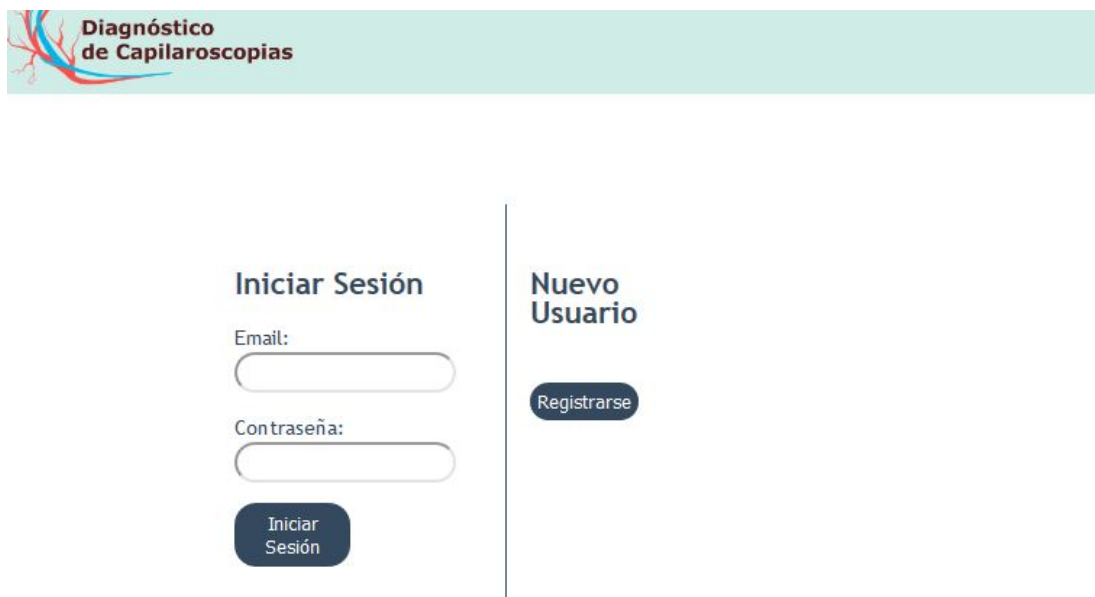
introducidos son válidos, se creará la cuenta y se volverá a la pantalla de inicio de sesión. En caso de que se viole alguna de las restricciones anteriores, se mostrará un mensaje explicando el error.

Una vez que el usuario se ha logueado, se pasará a la página de inicio (Figura 6.18), donde se mostrará una lista con todos los pacientes y una barra de búsqueda. Cada fila de la lista será un enlace para ver el perfil del paciente. También se muestra un botón para el registro de un nuevo paciente, que llevará a un formulario (Figura 6.19).

Al entrar en el perfil de un paciente (Figuras 6.20 y 6.21) se mostrará su información personal (y un botón para poder modificarla, que llevará a un formulario similar al de registro). También habrá tres apartados más: un apartado para subir las capilaroscopias, una sección de capilaroscopias que todavía no han sido diagnosticadas y otra sección con las que se han analizado.

Para diagnosticar una capilaroscopia, habrá que pulsar sobre ella. El sistema tardará unos tres segundos y luego mostrará la pantalla de diagnóstico (Figuras 6.22 y 6.23). En esta pantalla se mostrará la capilaroscopia, su información y una lista con los capilares analizados junto con su clasificación.

En todas las pantallas se dispone de un enlace para ir directamente al inicio, al perfil del usuario y para cerrar sesión. En el perfil del usuario (Figura 6.24) se mostrará la información personal de este, junto con dos botones: uno para modificar su nombre, apellidos y email (Figura 6.25) y otro para cambiar la contraseña (Figura 6.26).



The image shows a web interface for a medical application. At the top, there is a teal header bar with a logo on the left consisting of red and blue branching lines, and the text "Diagnóstico de Capilaroscopías" in white. Below the header, the page is divided into two columns by a vertical line. The left column is titled "Iniciar Sesión" and contains two input fields labeled "Email:" and "Contraseña:", each with a dark blue "Iniciar Sesión" button below them. The right column is titled "Nuevo Usuario" and contains a dark blue "Registrarse" button.

Diagnóstico de Capilaroscopías

Iniciar Sesión

Email:


Contraseña:

Iniciar Sesión

Nuevo Usuario

Registrarse

Figura 6.16: Pantalla de inicio de sesión.



Nuevo usuario

Email:

Nombre:

Apellidos:

DNI:

Contraseña:

Su contraseña no puede asemejarse tanto a su otra información personal.

Su contraseña debe contener al menos 8 caracteres.

Su contraseña no puede ser una clave utilizada comunmente.

Su contraseña no puede ser completamente numérica.

Contraseña (confirmación): Para verificar, introduzca la misma contraseña anterior.

[Registrarse](#)

Figura 6.17: Pantalla de registro de un usuario nuevo.



Diagnóstico de Capilaroscopias

Julia Gil Ordóñez

[Inicio](#) | [Mi perfil](#) | [Cerrar sesión](#)

Buscar por:

[Buscar](#) [Registrar paciente](#)

Pacientes

Nombre	Apellidos	Médico	Fecha de registro
Alan	Parsons	Julia Gil Ordóñez	20 de Agosto de 2019 a las 16:48
Antonio	Machado	Julia Gil Ordóñez	20 de Agosto de 2019 a las 21:01
Julia	Ordóñez	Julia Gil Ordóñez	21 de Agosto de 2019 a las 18:55

Figura 6.18: Pantalla de inicio.

The screenshot shows a web interface for registering a new patient. At the top, there is a navigation bar with links: Inicio | Mi perfil | Cerrar sesión. Below this, the title 'Nuevo Paciente' is centered. The form consists of several input fields: 'Nombre:', 'Apellidos:', 'DNI:', 'Teléfono:', and 'Email:', each followed by a text input box. Below these is a larger text area labeled 'Antecedentes médicos:'. At the bottom of the form is a 'Registrar' button. Below the button is a link '<< Volver a Inicio'.

Figura 6.19: Pantalla de registro de pacientes.

The screenshot shows a patient profile page. At the top, there is a navigation bar with links: inicio | mi perfil | Cerrar sesión. Below this, there is a link '<< Volver a inicio'. The main content area has a header with the patient's name 'Alan Parsons' and a 'Modificar perfil' button. Below this, there is a section titled 'Datos personales:' containing a table with the following information:

Nombre: Alan Parsons	Médico: Julia Gil Ordóñez
Teléfono: 981861044	Antecedentes médicos: No
Email: alan@parsons.com	

Below the table, there is a section titled 'Subir una nueva capilaroscopia para este paciente:'. It contains a button 'Examinar...' and a message 'No se ha seleccionado ningún archivo.' Below this is a 'Subir imagen' button.

Figura 6.20: Parte superior del perfil del paciente.

Imágenes sin diagnosticar

Para diagnosticar una imagen, pulse encima de ella. Una vez realizado el diagnóstico será redireccionado a la página de detalles de la capilaroscopia. El proceso puede tardar unos segundos.



Imágenes diagnosticadas

Puede ver los detalles de la capilaroscopia que desee pulsando encima de ella.

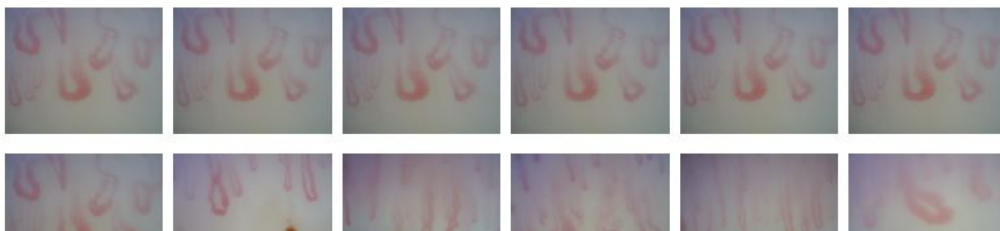



Figura 6.21: Parte inferior del perfil del paciente, donde se ven las capilaroscopias.

**Diagnóstico
de Capilaroscopias**

Julia Gil Ordóñez

Inicio | Mi perfil | Cerrar sesión

Alan Parsons

<< Volver al perfil del paciente



Capilares detectados: 6
Médico: Julia Gil Ordóñez
Fecha de diagnóstico: 23 de Agosto de 2019 a las 19:09

Figura 6.22: Parte superior de la pantalla de diagnóstico.

Detalle de los capilares

El número de capilares detectados puede variar con respecto a los capilares mostrados en detalle. Esto es debido a que los criterios para el conteo y la medición son diferentes. Para una estructura que esté formada por más de un capilar se mostrarán varias imágenes, sin embargo se contará como un capilar solo.

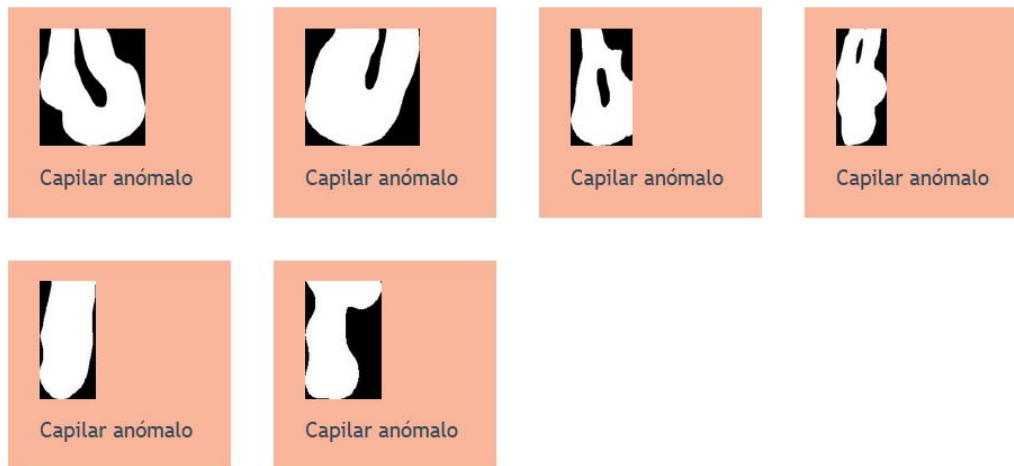


Figura 6.23: Parte inferior de la pantalla de diagnóstico.

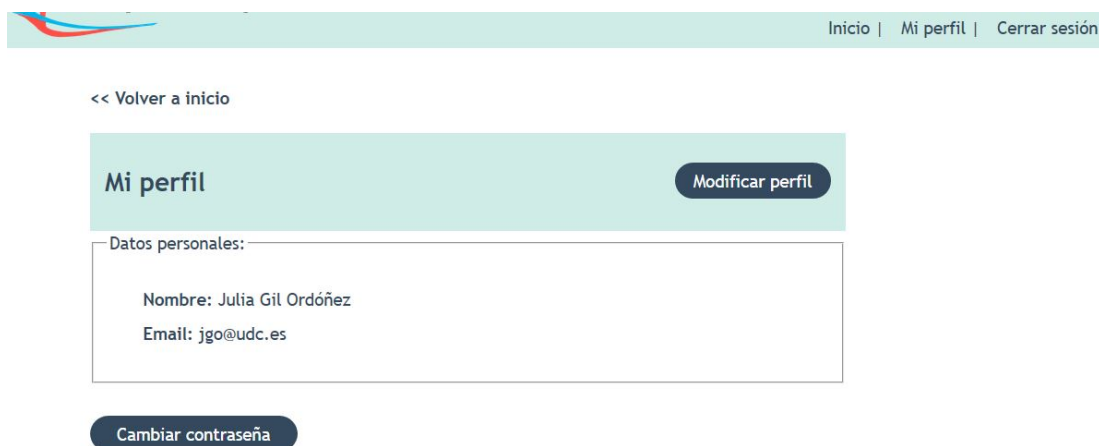


Figura 6.24: Pantalla de perfil de usuario.

<< Volver al perfil

Mi perfil

Datos personales:

Nombre:

Apellidos:

Email:

[Guardar cambios](#)

Figura 6.25: Pantalla de modificar el perfil de usuario.

<< Volver a Mi perfil

Cambio de contraseña

Contraseña antigua:

Contraseña nueva:

Su contraseña no puede asemejarse tanto a su otra información personal.
Su contraseña debe contener al menos 8 caracteres.
Su contraseña no puede ser una clave utilizada comunmente.
Su contraseña no puede ser completamente numérica.

Contraseña nueva (confirmación):

[Guardar cambios](#)

Figura 6.26: Pantalla de modificar la contraseña del usuario.

6.4 Pruebas

Se han efectuado una serie de pruebas unitarias para comprobar que el modelo de la aplicación funciona correctamente. Django posee un archivo específico para los casos de prueba (**tests.py**), por lo que será en el que se incluyan todos ellos. En la Figura 6.27 se puede ver un diagrama con la estructura de este archivo. Contiene tres clases: en **UserTests** están los tests para Usuario, **PacienteTests** contiene los de Paciente y en **CapilaroscopiaTests** se encuentran los tests de Capilaroscopia y Capilar. En conjunto hacen un total de 29 casos de prueba.

UserTests	PacienteTests	CapilaroscopiaTests
<pre> +setUpTestData() +test_first_name_get() +test_last_name_get() +test_email_get() +test_dni_get() +test_first_name_update() +test_last_name_update() +test_email_update() +test_invalid_email() +test_duplicated_email() +test_duplicated_dni() </pre>	<pre> +setUpTestData() +test_patient_first_name_get() +test_patient_last_name_get() +test_patient_email_get() +test_patient_phone_get() +test_patient_dni_get() +test_patient_doctor_get() +test_patient_first_name_update() +test_patient_last_name_update() +test_patient_email_update() +test_patient_phone_update() +test_patient_invalid_email() +test_patient_duplicated_email() +test_patient_duplicated_dni() </pre>	<pre> +setUpTestData() +test_user_get() +test_patient_get() +test_count_get() +test_capillar_capillaroscopia_get() +test_capillar_diagnosis_get() +test_add_capillars() </pre>

Figura 6.27: Tests efectuados para el modelo de la aplicación.

Conclusiones

Una capilaroscopia es una técnica de diagnóstico que estudia los capilares sanguíneos de manera no invasiva. Se aplica sobre el lecho ungueal y obtiene imágenes microscópicas de los vasos allí situados. Se utiliza para la detección de enfermedades vasculares como la esclerodermia sistémica o el lupus eritematoso.

El objetivo de este proyecto era la creación de una herramienta automática que fuese capaz de analizar y diagnosticar imágenes de capilaroscopia. De esta manera se facilitaría la tarea a los expertos, cuyo proceso de análisis manual resultaba tedioso y se alargaba demasiado.

Para el cumplimiento de este objetivo, se planificó el proyecto haciendo uso de una metodología ágil basada en Scrum y se dividió en seis *sprints*.

En primer lugar, fue necesario realizar una criba de las imágenes de las que se disponía para escoger aquellas que no tuviesen defectos graves de adquisición, es decir, que no hubiera nada en ellas que pudiera entorpecer el visionado de los vasos, como burbujas o reflejos. Una vez obtenidas las imágenes, se procedió a crear de forma manual máscaras binarias que indicaran la localización de los vasos en las capilaroscopias. Estas servirían de guía para el entrenamiento de una red de neuronas convolucional de tipo U-Net.

Para que la segmentación generada por la U-Net fuese más precisa, se creó un algoritmo de preprocesado de las imágenes que aumentaba su contraste y hacía más visibles los capilares. Se generaron varios posibles modelos de segmentación, modificando tanto parámetros de la red como las características del conjunto de entrenamiento. El mejor modelo, entrenado con las imágenes originales preprocesadas, obtuvo un 93.63 % de *accuracy*.

A continuación se procedió a diseñar un método para realizar un conteo automático de los vasos presentes en cada capilaroscopia. Para ello se recortaba la imagen para encuadrar la zona externa de los capilares y se obtenían los componentes conexos más grandes. Se volvía a recortar cada uno de los componentes para obtener las cabezas de capilar que pudieran estar unidas por el cuerpo y se hacía un recuento de todas las cabezas detectadas. El sistema coincidió en un 90.45 % con un conteo manual de los capilares.

Desarrollada la metodología de conteo, se procedió a crear un método para la clasificación de cada capilar en *sano* o *anómalo*. Se utilizaron tres parámetros diferentes para determinar a qué clase pertenecían: se tuvo en cuenta su longitud, su anchura y su regularidad. Si el capilar analizado era largo, fino y regular, sería considerado *sano*. Se obtuvo un 90.73 % de coincidencias con respecto a una clasificación manual de las estructuras.

Finalmente se procedió a la creación de la aplicación web. Para esto, primero se identificaron los requisitos del sistema. Luego se hizo un diseño previo, donde se incluyeron los patrones que implementaría la aplicación, el diseño de la base de datos y las maquetas de todas las pantallas. Posteriormente se realizó la implementación haciendo uso del framework Django de Python.

Tras la realización del trabajo, se puede concluir que los objetivos propuestos se han cumplido en su totalidad, alcanzando además unos porcentajes de acierto en la detección y clasificación de capilares bastante buenos. Aun así, existen una serie de mejoras que se podrían aplicar como futuras líneas de trabajo tales como:

- Entrenamiento de la red con recortes de la imagen original para tratar de mejorar la segmentación de los detalles de los capilares.
- Mejora del proceso de conteo.
- Diagnóstico de varias imágenes simultáneas para agilizar el proceso en lotes grandes de capilaroscopias.
- Ejecución asíncrona de diagnóstico de imágenes para mejorar la experiencia de usuario.
- Internacionalización de la aplicación web.
- Uso de librerías javascript para subir múltiples imágenes desde el equipo del usuario al navegador.

Apéndice

Glosario de acrónimos

CNN *Convolutional Neural Network.*

RNA *Red de neuronas artificiales.*

TFG *Trabajo de Fin de Grado.*

Glosario de términos

Etiología Causas que originan una enfermedad.

Capilaroscopia Técnica de diagnóstico que estudia los capilares sanguíneos de manera no invasiva.

Lecho ungueal Región de las uñas y su periferia.

Esclerodermia sistémica Enfermedad crónica caracterizada por cambios degenerativos en la piel, las articulaciones y los órganos internos.

Síndrome de Raynaud Afección vascular que afecta a arterias pequeñas, en especial a las de las manos y pies.

Bibliografía

- [1] V. Garra, N. Danese, M. Rebella, and E. e. Cairoli, “Capilaroscopia en el diagnóstico de las enfermedades autoinmunes sistémicas,” *Revista Médica del Uruguay*, vol. 28(2), pp. 89–98, 2012.
- [2] R. A. Hajj-ali, “Esclerosis sistémica (versión para público general),” [Accessible el 26 de junio de 2019]. [Online]. Available: <https://www.msdmanuals.com/es-es/hogar/trastornos-de-los-huesos,-articulaciones-y-m%C3%BAsculos/enfermedades-autoinmunitarias-del-tejido-conjuntivo/esclerosis-sist%C3%A9mica>
- [3] J. W. Hallett, Jr., “Síndrome de raynaud (versión para público general),” [Accessible el 26 de junio de 2019]. [Online]. Available: <https://www.msdmanuals.com/es-es/hogar/trastornos-del-coraz%C3%B3n-y-los-vasos-sangu%C3%ADneos/arteriopat%C3%ADa-da-perif%C3%A9rica/s%C3%ADndrome-de-raynaud>
- [4] S. G. Sheps, “Ecografía doppler,” [Accessible el 30 de junio de 2019]. [Online]. Available: <https://www.mayoclinic.org/es-es/doppler-ultrasound/expert-answers/faq-20058452>
- [5] H. Corominas and V. Ortiz-Santamaria, “Técnica capilaroscópica,” *Elsevier Doyma*, vol. 11 (Supl. 1), pp. 8–10, 2010.
- [6] O. L. Sedano Santiago, J. Sedano Barreto, L. Ventura-Ríos, and M. Gutiérrez, “Perspectivas en el estudio del fenómeno de raynaud desde un punto de vista ecográfico,” *Revista Chilena de Reumatología*, vol. 28(4), pp. 194–199, 2012.
- [7] Springer Science and Business Media, “Síndrome de raynaud con palidez,” [Accessible el 30 de junio de 2019]. [Online]. Available: https://www.msdmanuals.com/-/media/manual/home/images/42_raynaud_slide_4_springer_high_es.jpg?la=es&thn=0
- [8] Hospital Cima de Barcelona, “Capilaroscopia,” [Accessible el 27 de junio de 2019]. [Online]. Available: <https://www.hospitalcima.es/es/pruebas-diagnosticas-capilaroscopia.php>

-
- [9] J. P. Restrepo, M. Gutiérrez R., R. De Angelis, and W. Grassi, “El papel de la capilaroscopia del lecho ungueal en reumatología,” *Asociación Colombiana de Reumatología*, vol. 15(3), pp. 187–195, 2008.
- [10] M. Berks, P. Tresaden, G. Dinsdale, A. Murray, T. Moore, C. Taylor, and A. L. Herrick, “An Automated System for Detecting and Measuring Nailfold Capillaries,” *Med Image Comput Comput Assist Interv.*, vol. 17 (1), pp. 658–665, 2014.
- [11] M. Berks, G. Dinsdale, A. Murray, T. Moore, J. Manning, C. Taylor, and A. L. Herrick, “Automated structure and flow measurement — a promising tool in nailfold capillaroscopy,” *Microvascular Research*, vol. 118, pp. 173–177, 2018.
- [12] C. Cheng, C. W. Lee, and C. Daskalakis, “A Reproducible Computerized Method for Quantitation of Capillary Density using Nailfold Capillaroscopy,” *Journal of Visualized Experiments*, vol. 104, p. 53088, 2015.
- [13] H. Kwasnicka and M. Paradowski, “Capillaroscopy image analysis as an automatic image annotation problem,” 07 2007, pp. 266–271.
- [14] W3Schools, “Introducción a python,” [Accessible el 2 de julio de 2019]. [Online]. Available: <https://proyectosagiles.org/que-es-scrum/>
- [15] Proyectos ágiles, “Qué es scrum,” [Accessible el 20 de agosto de 2019]. [Online]. Available: https://www.w3schools.com/python/python_intro.asp
- [16] Numpy developers, “Numpy,” accessible el 2 de julio de 2019]. [Online]. Available: <https://www.numpy.org/>
- [17] F. Lundh et al., “Python imaging library (pil),” accessible el 28 de julio de 2019]. [Online]. Available: <https://www.pythonware.com/products/pil/>
- [18] Google Inc., “Tensorflow homepage,” [Accessible el 2 de julio de 2019]. [Online]. Available: <https://www.sqlite.org/about.html>
- [19] OpenCV Team, “Opencv homepage,” [Accessible el 4 de julio de 2019]. [Online]. Available: <https://opencv.org/>
- [20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [21] Django Software Foundation, “Django homepage,” [Accessible el 6 de julio de 2019]. [Online]. Available: <https://www.djangoproject.com/>

- [22] SQLite Consortium, “About sqlite,” [Accessible el 2 de julio de 2019]. [Online]. Available: <https://www.sqlite.org/about.html>
- [23] P. Raybaut, C. Cordoba et al., “Spyder website,” [Accessible el 24 de julio de 2019]. [Online]. Available: <https://www.spyder-ide.org/>
- [24] G. Alder, D. Benson, “About draw.io,” [Accessible el 24 de julio de 2019]. [Online]. Available: <https://about.draw.io/support/>
- [25] Adobe Systems Inc., “Adobe photoshop,” [Accessible el 25 de julio de 2019]. [Online]. Available: <https://www.adobe.com/Photoshop>
- [26] R. Keith-Magee et al., “Github homepage,” [Accessible el 25 de julio de 2019]. [Online]. Available: <https://github.com/>
- [27] E. Posner et al., “Taiga,” [Accessible el 24 de julio de 2019]. [Online]. Available: <https://taiga.io>
- [28] F. José Palacios Burgos, “La neurona artificial,” [Accessible el 3 de agosto de 2019]. [Online]. Available: https://www.ibiblio.org/pub/linux/docs/LuCaS/Presentaciones/200304curso-glisa/redes_neuronales/curso-glisa-redes_neuronales-html/x38.html
- [29] S. Saha, “A comprehensive guide to convolutional neural networks — the eli5 way,” [Accessible el 5 de agosto de 2019]. [Online]. Available: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- [30] F. Giménez-Palomares, J. A. Monsoriu and E. Alemany-Martínez, “Aplicación de la convolución de matrices al filtrado de imágenes,” *Modelling in Science Education and Learning*, vol. 9(2), 2016.
- [31] R. E. Lopez Briega, “Redes neuronales convolucionales con TensorFlow,” [Accessible el 7 de agosto de 2019]. [Online]. Available: <https://relopezbriega.github.io/blog/2016/08/02/redes-neuronales-convolucionales-con-tensorflow/>
- [32] D. Calvo, “Redes neuronales convolucionales con TensorFlow,” [Accessible el 7 de agosto de 2019]. [Online]. Available: <http://www.diegocalvo.es/red-neuronal-convolucional/>
- [33] NeuroHive, “U-Net: Image Segmentation Network,” [Accessible el 7 de agosto de 2019]. [Online]. Available: <https://neurohive.io/en/popular-networks/u-net/>
- [34] OpenCV Team, “Changing the contrast and brightness of an image! ,” [Accessible el 12 de agosto de 2019]. [Online]. Available: https://docs.opencv.org/master/d3/dc1/tutorial_basic_linear_transform.html

- [35] F. G. Loch, “Image Processing Algorithms Part 5: Contrast Adjustment,” [Accessible el 12 de agosto de 2019]. [Online]. Available: <https://www.dfstudios.co.uk/articles/programming/image-programming-algorithms/image-processing-algorithms-part-5-contrast-adjustment/>
- [36] Machine Learning Cheatsheet, “Loss functions,” [Accessible el 11 de julio de 2019]. [Online]. Available: https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions
- [37] F. Chollet, “Keras source code,” [Accessible el 28 de agosto de 2019]. [Online]. Available: https://github.com/keras-team/keras-preprocessing/blob/master/keras_preprocessing/image/utils.py
- [38] F. Lundh et al., “Pillow image module,” [Accessible el 28 de agosto de 2019]. [Online]. Available: <https://pillow.readthedocs.io/en/3.1.x/reference/Image.html>
- [39] U. Sinha, “Image Moments,” [Accessible el 2 de agosto de 2019]. [Online]. Available: <http://aishack.in/tutorials/image-moments/>
- [40] U. Hernández, “MVC (Model, View, Controller) explicado,” [Accessible el 10 de agosto de 2019]. [Online]. Available: <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>
- [41] TutorialsPoint, “Design Pattern - Singleton Pattern,” [Accessible el 10 de agosto de 2019]. [Online]. Available: https://www.tutorialspoint.com/design_pattern/singleton_pattern